



TESIS - TE142599

**PENGENALAN HURUF BRAILLE MENGGUNAKAN
METODE BLOB ANALYSIS DAN ARTIFICIAL
NEURAL NETWORK**

**JOKO SUBUR
2213204009**

**DOSEN PEMBIMBING
Dr. Tri Arief Sardjono, ST., MT.
Ronny Mardiyanto, ST., MT., Ph.D.**

**PROGRAM MAGISTER
BIDANG KEAHLIAN TEKNIK ELEKTRONIKA
JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2015**



THESIS - TE142599

BRaille CHARACTER RECOGNITION USING BLOB ANALYSIS AND ARTIFICIAL NEURAL NETWORK METHOD

**JOKO SUBUR
2213204009**

**SUPERVISOR
Dr. Tri Arief Sardjono, ST., MT.
Ronny Mardiyanto, ST., MT., Ph.D.**

**MAGISTER PROGRAM
FIELD IN ELECTRONICS
ELECTRICAL DEPARTMENT
FACULTY OF INDUSTRIAL TECHNOLOGY
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2015**

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar
Magister Teknik (M.T)

di

Institut Teknologi Sepuluh Nopember

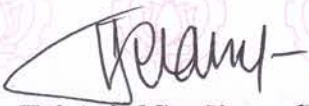
Oleh:


Joko Subur


Nrp. 2213204009


Tanggal Ujian : 23 Juni 2015
Periode Wisuda : September 2015

Disetujui oleh:


1. Dr. Tri Arief Sardjono, ST., MT. (Pembimbing I)
NIP: 19700212 199512 1 001


2. Ronny Mardiyanto, ST., MT., Ph.D. (Pembimbing II)
NIP: 19810118 200312 1 003


3. Ir. Djoko Purwanto, M.Eng., Ph.D. (Penguji)
NIP: 19651211 199002 1 002


4. Dr. Muhammad Rivai, ST., MT. (Penguji)
NIP: 19690426 199403 1 003


5. Achmad Arifin, ST., M.Eng., Ph.D. (Penguji)
NIP: 19710314 199702 1 001



Direktur Program Pascasarjana,

Prof. Dr. Ir. Adi Soeprijanto, MT.
NIP: 19640405 199002 1 001

PENGENALAN HURUF BRAILLE MENGGUNAKAN METODE BLOB ANALYSIS DAN ARTIFICIAL NEURAL NETWORK

Nama Mahasiswa : Joko Subur

NRP : 2213204009

Pembimbing : Dr. Tri Arief Sardjono, ST., MT.
Ronny Mardiyanto, ST., MT., Ph.D.

ABSTRAK

Huruf braille merupakan jenis huruf yang didesain khusus bagi tuna netra, tersusun dari enam titik timbul. Enam titik tersebut disusun sedemikian rupa sehingga menciptakan bermacam kombinasi. Pada umumnya huruf braille di baca dengan cara diraba dengan telapak tangan, oleh karena itu diperlukan kepekaan telapak tangan terhadap titik-titik timbul dan harus mengerti serta hafal kombinasi titik-titik timbul tersebut dalam membentuk suatu huruf. Sehingga tidak semua orang bisa membaca huruf braille, kebanyakan masih kesulitan dalam membaca huruf braille. Penelitian ini bertujuan untuk membuat sistem pendeteksi dan penterjemah karakter braille ke huruf abjad. Digunakan camera webcam untuk pengambilan gambar huruf braille. Dari gambar huruf braille kemudian diproses pengolahan citra melalui tahapan cropping, grayscale, thresholding, erosion, dilation, blob analysis dan pengenalan gambar braille menggunakan artificial neural network. Hasil penelitian ini sistem dapat mengenali karakter braille dan menterjemahkannya ke bentuk huruf abjad dengan tingkat akurasi 99%.

Kata kunci: Artificial neural network, Blob Analysis, Huruf braille, Pengolahan citra, Webcam.

(Halaman ini sengaja dikosongkan)

BRaille CHARACTER RECOGNITION USING BLOB ANALYSIS AND ARTIFICIAL NEURAL NETWORK METHOD

By : Joko Subur
Student Identity Number : 2213204009
Supervisor : Dr. Tri Arief Sardjono, ST., MT.
Ronny Mardiyanto, ST., MT., Ph.D.

ABSTRACT

Braille letter is characters designed for the blind, consist of six embossed points, arranged in a standard braille character. Braille letters is touched and read using fingers, therefore the sensitivity of the fingers is the key issue here. Those characters need to be memorized, so it is very difficult to be learned. The aim of this research is to create a braille characters recognition system and translate it to alpha-numeric text. Webcam camera is used to capture braille image from braille characters on the paper sheet. Cropping, grayscale, thresholding, erosion, and dilation techniques are used for image preprocessing. Then, blob analysis and artificial neural network method are used to recognize the braille characters. The system can recognize braille characters with 99%.

Keywords : Artificial neural network, Blob analysis, Braille character, Image processing, Webcam.

(Halaman ini sengaja dikosongkan)

KATA PENGANTAR

Alhamdulillah, puji dan syukur penulis panjatkan kepada Allah SWT, karena atas segala nikmat-Nya tesis ini dapat diselesaikan. Tesis berjudul *“Pengenalan Huruf Braille Menggunakan Metode Blob Analysis Dan Artificial Neural Network”* ini disusun untuk memenuhi sebagian persyaratan memperoleh gelar Magister Teknik (MT) pada Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember.

Penulis menyadari bahwa dalam penyusunan tesis ini tidak terlepas dari bantuan berbagai pihak. Oleh karena itu, dengan ketulusan dan kerendahan hati penulis menyampaikan terima kasih kepada:

1. Bapak Dr. Tri Arief Sardjono, ST., MT selaku dosen pembimbing I yang telah banyak memberikan saran, bantuan serta bimbingan dan selaku Ketua Jurusan Teknik Elektro.
2. Bapak Ronny Mardiyanto, ST., MT., Ph.D selaku dosen pembimbing II yang telah banyak memberikan saran, bantuan serta bimbingan.
3. Bapak Ir. Djoko Purwanto, M.Eng., Ph.D selaku koordinator program Pasca Sarjana jurusan Teknik Elektro.
4. Bapak Achmad Arifin, ST., M.Eng., Ph.D selaku koordinator Bidang Studi Elektronika.
5. Bapak Dr. Muhammad Rivai, ST., MT Selaku dosen pengajar Jurusan Teknik Elektro yang telah banyak memberikan ilmu selama penulis menempuh kuliah.
6. Ayahanda atas segala dukungan dan doanya hingga sampai terselesaikannya tesis ini.
7. Bapak dan Ibu Mertua atas segala dukungan dan doanya hingga sampai terselesaikannya tesis ini.
8. Istriku tercinta Wiwin Suliasih yang senantiasa memberikan dukungan dan doanya.
9. Cahaya Hati Ayah Adek Aisyah Aqilah Sekar Ayu.

10. Rekan–rekan seperjuangan angkatan 2013, mas Adi, mas Hadid, mas Rendy, mas Dedi, Mas Roy, mas Bagus, mas Bakti, mas Rizal, mas Tama, mas Wahyu, mas Sulfan, mas Rahmad, mbak Yanti, mbak Eva, mbak Nada, mbak Putri, mbak Dina, mbak Nita serta teman-teman mahasiswa bidang studi elektronika atas segala bantuan dan sumbangan pikiran dan tenaga dalam menyelesaikan tesis ini.
11. Seluruh Civitas Akademisi Jurusan Teknik Elektro Fakultas Teknologi Industri Institut Teknologi Sepuluh Nopember Surabaya., atas segala bantuan yang telah diberikan.

Semoga Allah SWT membalas budi baik mereka semua.

Yang terakhir Special untuk Almarhumah Ibunda tercinta, Semoga beliau tetap tersenyum di SURGA sana, Amin.

“ Putramu telah mewujudkan cita-citamu IBU.., I Love You Mom... “

Pada akhirnya, penulis menyadari bahwa tesis ini masih belum sempurna. Oleh karena itu, penulis sangat mengharapkan kritik dan saran yang membangun. Penulis berharap semoga tesis ini dapat bermanfaat bagi pengembangan ilmu pengetahuan dan teknologi serta bagi masyarakat demi kemajuan Bangsa.

Terima kasih.

Surabaya, 10 Juli 2015

Penulis

DAFTAR ISI

COVER	
PERNYATAAN KEASLIAN TESIS	
LEMBAR PENGESAHAN BUKU TESIS	
ABSTRAK	i
ABSTRACT	iii
KATA PENGANTAR	v
DAFTAR ISI	vii
DAFTAR GAMBAR	xi
DAFTAR TABEL	xv
BAB 1 PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Perumusan Masalah.....	2
1.3. Batasan Masalah	3
1.3. Tujuan dan Manfaat Penelitian	3
BAB 2 KAJIAN PUSTAKA DAN DASAR TEORI	5
2.1. Kajian Pustaka	5
2.1.1. Pengenalan Huruf Braille Metode Neural Network for Half Character.....	5
2.1.2. Pengenalan Huruf Braille Metode Cellular Neural Network.....	7
2.1.3. Pengenalan Huruf Braille Metode Support Vector Machine Classifier	8
2.1.4. Pengenalan Huruf Braille Metode Optical Braille Recognition	9
2.1.5. Fishbone Diagram	13
2.2. Dasar Teori	14
2.2.1. Huruf Braille	14
2.2.2. Pengolahan Citra	15
2.2.3. Artificial Neural Network.....	21
BAB 3 METODE PENELITIAN	29
3.1. Pengambilan Gambar Braille.....	30

3.2.	Pengolahan Gambar Braille	33
3.2.1.	Crop Image.....	33
3.2.2.	Proses Grayscale	35
3.2.3.	Proses Thresholding	36
3.2.4.	Proses Dilatasi.....	38
3.2.5.	Proses Erosi.....	39
3.3.	Pembacaan Huruf Braille.....	41
3.3.1.	Blob Analysis	41
3.3.2.	Artificial Neural Network.....	50
BAB 4 PENGUJIAN DAN ANALISA SISTEM.....		69
4.1.	Pengujian Pembacaan Jumlah Titik Dot Hitam	69
4.2.	Pengujian Pembacaan Koordinat Setiap Titik Dot Hitam	70
4.3.	Pengujian Penentuan Area Segmentasi Setiap Huruf Braille	72
4.4.	Pengujian Pengenalan Huruf Braille di Terjemahkan ke Teks	74
4.4.1.	Pengujian Pengenalan Huruf Braille Pada Cetakan Tanpa Spasi ...	74
4.4.2.	Pengujian Pengenalan Huruf Braille Pada Cetakan Jarak Satu Spasi	75
4.4.3.	Pengujian Pengenalan Huruf Braille Pada Cetakan Jarak Dua Spasi	76
4.4.4.	Pengujian Pengenalan Huruf Braille Pada Cetakan Jarak Tiga Spasi	78
4.4.5.	Pengujian Pengenalan Huruf Braille Pada Cetakan Buku Cerita	79
4.4.6.	Pengujian Pengenalan Huruf Braille Pada Cetakan Printer Braillo-400 (Lab. A304 Teknik Elektro – ITS)	80
4.5.	Pengujian Pengenalan Huruf Braille Dengan Tingkat Kemiringan Pada Gambar Dari 0 Derajat Sampai 1,5 Derajat	81
4.6.	Pengujian Pengenalan Huruf Braille Dengan Tingkat Kemiringan Sumber Cahaya Terhadap Kertas Braille	82
4.7.	Pengujian Pengenalan Huruf Braille Dengan Pengambilan Gambar Menggunakan Alat Scanner	88
4.8.	Pengujian Pengenalan Huruf Braille Pada Cetakan Double Side (2 sisi) Pengambilan Gambar dengan Kamera Webcam	90
4.9.	Pengujian Sistem Pengenalan Huruf Braille Pada Cetakan Double Side (2 sisi) Pengambilan Gambar dengan Alat Scanner.....	93

BAB 5 PENUTUP	97
5.1. Kesimpulan dan Saran.....	97
5.2. Saran.....	97
DAFTAR PUSTAKA	99
LAMPIRAN-LAMPIRAN	
BIODATA PENULIS	

(Halaman ini sengaja dikosongkan)

DAFTAR TABEL

Tabel 2.1 Matriks Dari Cell t_{45}	8
Tabel 3.1 Data Hasil Pengambilan Data Input Dengan Kemiringan Gambar 0 derajat (tegak lurus)	53
Tabel 3.2 Data Hasil Pengambilan Data Input Dengan Kemiringan Gambar 0,5 (setengah) derajat	55
Tabel 3.3 Data Hasil Pengambilan Data Input Dengan Kemiringan Gambar 1 derajat	57
Tabel 3.4 Data Hasil Pengambilan Data Input Dengan Kemiringan Gambar -0,5 derajat	59
Tabel 3.5 Data Hasil Pengambilan Data Input Dengan Kemiringan Gambar -1 derajat	61
Tabel 3.6 Lookup Tabel Data Keluaran Proses ANN Setiap Huruf	68
Tabel 4.1 Data Hasil Pengujian Pembacaan Jumlah Titik Dot Hitam	70
Tabel 4.2 Data Pengujian Pengenalan Huruf Braille Cetakan Tanpa Spasi	75
Tabel 4.3 Data Pengujian Pengenalan Huruf Braille Cetakan Jarak Satu Spasi ...	76
Tabel 4.4 Data Pengujian Pengenalan Huruf Braille Cetakan Jarak Dua Spasi...	77
Tabel 4.5 Data Pengujian Pengenalan Huruf Braille Cetakan jarak Tiga Spasi ...	78
Tabel 4.6 Data Pengujian Pengenalan Huruf Braille Pada Cetakan Buku Cerita Yang Berjudul “Hua Lo Puu”	79
Tabel 4.7 Data Pengujian Pengenalan Huruf Braille Hasil Cetakan Printer Braillo-400 (di Lab. Elektro ITS)	80
Tabel 4.8 Data Hasil Pengujian Pembacaan Huruf Braille dengan Kemiringan Gambar Braille 0 Derajat – 1,5 Derajat	81
Tabel 4.9 Data Hasil Pengujian Pembacaan Huruf Braille dengan Kemiringan Sumber Cahaya Terhadap Kertas Braille	87
Tabel 4.10 Data Hasil Pengujian Pembacaan Huruf Braille Untuk Gambar Dari Hasil Alat Scanner Cetakan 1 Sisi	89
Tabel 4.11 Data Hasil Pengujian Pembacaan Huruf Braille Untuk Gambar Dari Hasil Kamera Webcam Pada Cetakan 2 Sisi	92

Tabel 4.11 Data Hasil Pengujian Pembacaan Huruf Braille Untuk Gambar Dari Hasil Scanner Pada Cetakan 2 Sisi	95
--	----

DAFTAR GAMBAR

Gambar 2.1 Hasil Proses Scanning Huruf Braille dengan Alat Scanner	5
Gambar 2.2 Diagram Alir Sistem Pengenalan Huruf Braille Metode Neural Network for Half-Character	6
Gambar 2.3 Struktur Cell pada CNN	7
Gambar 2.4 Penentuan Nilai Segmen Cell pada CNN	8
Gambar 2.5 Hasil Proses Scanning Huruf Braille Cetakan Dua Sisi	9
Gambar 2.6 Diagram Sistem Pengenalan Huruf Braille Metode Support Vector Machine	9
Gambar 2.7 Konversi Citra Grayscale Menjadi Citra Biner Menggunakan Haar Feature dan Support Vector Machine	10
Gambar 2.8 Tiga Jenis Tipe Wavelet Haar pada Metode SVM	10
Gambar 2.9 Hasil Proses Citra Biner dengan SVM	11
Gambar 2.10 Diagram Sistem Pengenalan Huruf Braille Metode OBR	12
Gambar 2.11 Gambar Posisi Miring dan Gambar Posisi Lurus	12
Gambar 2.12 Hasil Proses Segmentasi dan Hasil Perhitungan dari Hasil Segmentasi	13
Gambar 2.13 Fishbone Diagram Penelitian Sebelumnya	13
Gambar 2.14 Susunan Titik-titik Dot pada Huruf Braille	14
Gambar 2.15 Susunan Kombinasi Titik-titik Dot pada Huruf Braille	14
Gambar 2.16 Membaca Huruf Braille dengan Diraba	15
Gambar 2.17 Structure Element dan Gambar yang akan Diproses Dilatasi	17
Gambar 2.18 Ilustrasi Proses Dilatasi	18
Gambar 2.19 Structure Element dan Gambar yang akan Diproses Erosi	18
Gambar 2.20 Ilustrasi Proses Erosi	18
Gambar 2.21 Hubungan Garis Lurus Membentuk Prediksi Garis Tepi	19
Gambar 2.22 Kode Rantai Arah 4 Mata Angin dan 8 Mata Angin	20
Gambar 2.23 Arsitektur Artificial Neural Network	21
Gambar 2.24 Fungsi Sigmoid Biner	24
Gambar 2.25 Fungsi Sigmoid Bipolar	24

Gambar 3.1 Diagram Blok Proses Pembacaan Huruf Braille Menggunakan Metode Blob Analysis dan Artificial Neural Network.....	29
Gambar 3.2 Ilustrasi Teknik Pencahayaannya Pada Proses Pengambilan Gambar Braille	30
Gambar 3.3 Ilustrasi Teknik Pengambilan Gambar Braille Pada Kertas Braille dengan 2 Webcam dan 3 Posisi.....	31
Gambar 3.4 Ilustrasi Kontrol Posisi Webcam Dengan Motor Stepper	31
Gambar 3.5 Contoh Hasil Pengambilan Gambar Braille Dengan Webcam	33
Gambar 3.6 Hasil Proses Cropping Image	34
Gambar 3.7 Diagram Alir Proses Grayscale	35
Gambar 3.8 Gambar RGB dan Gambar Hasil Proses Grayscale.....	36
Gambar 3.9 Hasil Adaptive Threshold dan Hasil Threshold Biasa.....	36
Gambar 3.10 Diagram Alir Proses Thresholding	37
Gambar 3.11 Hasil Proses Adaptive Threshold (masih terdapat noise)	38
Gambar 3.12 Hasil Proses Dilatasi (noise hilang)	38
Gambar 3.13 Diagram Alir Proses Dilatasi.....	39
Gambar 3.14 Hasil Proses Erosi (Titik Hitam Membesar Kembali)	40
Gambar 3.15 Diagram Alir Proses Erosi.....	40
Gambar 3.16 Representasi Proses Blob Analysis.....	42
Gambar 3.17 Diagram Alir Proses Blob Analysis.....	43
Gambar 3.18 Hasil Pembacaan Titik Tengah Setiap Dot Hitam.....	44
Gambar 3.19 Diagram Alir Proses Mencari Titik Tengah Koordinat x.....	45
Gambar 3.20 Diagram Alir Proses Mencari Titik Tengah Koordinat y.....	46
Gambar 3.21 Hasil Mencari Koordinat Rata-rata Setiap Titik Dot Hitam	48
Gambar 3.22 Hasil Proses Segmentasi Area	49
Gambar 3.23 Segmentasi Kecil Pada Area Satu Huruf Braille	49
Gambar 3.24 Topologi Jaringan Artificial Neural Network Pada Sistem Pengenalan Huruf Braille.....	50
Gambar 3.25 Area Segmentasi Kecil Pada Area Satu Huruf Braille.....	52
Gambar 3.26 Pengambilan Data Input Pada Kemiringan Gambar 0 Derajat	52
Gambar 3.27 Pengambilan Data Input Pada Kemiringan Gambar 0,5 Derajat	54
Gambar 3.28 Pengambilan Data Input Pada Kemiringan Gambar 1 Derajat	56

Gambar 3.29 Pengambilan Data Input Pada Kemiringan Gambar -0,5 Derajat ..	58
Gambar 3.30 Pengambilan Data Input Pada Kemiringan Gambar -1 Derajat	60
Gambar 4.1 Hasil Pengujian Pembacaan Koordinat Setiap Titik Dot Hitam	72
Gambar 4.2 Penentuan Area Segmentasi Pada Setiap Huruf Braille	73
Gambar 4.3 Segmentasi Pada Tiap Huruf Braille Dijadikan 40 Area Kecil.....	73
Gambar 4.4 Cetakan Huruf Braille Tanpa Jarak Spasi	74
Gambar 4.5 Cetakan Huruf Braille Dengan Jarak Satu Spasi	75
Gambar 4.6 Cetakan Huruf Braille Dengan Jarak Dua Spasi	77
Gambar 4.7 Cetakan Huruf Braille Dengan Jarak Tiga Spasi.....	78
Gambar 4.8 Ilustrasi Penempatan Sumber Pencahayaan Dengan Kemiringan 1 Derajat Terhadap Kertas Braille	82
Gambar 4.9 Hasil Dengan Sumber Pencahayaan kemiringan 1 Derajat	78
Gambar 4.8 Ilustrasi Penempatan Sumber Pencahayaan Dengan Kemiringan 1 Derajat Terhadap Kertas Braille	82
Gambar 4.9 Hasil Dengan Sumber Pencahayaan kemiringan 1 Derajat	83
Gambar 4.10 Ilustrasi Penempatan Sumber Pencahayaan Dengan Kemiringan 5 Derajat Terhadap Kertas Braille	83
Gambar 4.11 Hasil Dengan Sumber Pencahayaan kemiringan 5 Derajat.....	83
Gambar 4.12 Ilustrasi Penempatan Sumber Pencahayaan Dengan Kemiringan 20 Derajat Terhadap Kertas Braille	84
Gambar 4.13 Hasil Dengan Sumber Pencahayaan kemiringan 20 Derajat.....	84
Gambar 4.14 Ilustrasi Penempatan Sumber Pencahayaan Dengan Kemiringan 45 Derajat Terhadap Kertas Braille	85
Gambar 4.15 Hasil Dengan Sumber Pencahayaan kemiringan 45 Derajat.....	85
Gambar 4.16 Ilustrasi Penempatan Sumber Pencahayaan Dengan Kemiringan 90 Derajat Terhadap Kertas Braille	86
Gambar 4.17 Hasil Dengan Sumber Pencahayaan kemiringan 90 Derajat.....	86
Gambar 4.18 Hasil Pengambilan Gambar Braille Cetakan 1 Sisi Dengan Alat Scanner.....	88
Gambar 4.19 Hasil General Threshold dan Adaptive Threshold Pada Gambar Braille Cetakan 1 Sisi Dari Scanner	89
Gambar 4.20 Hasil Pengambilan Gambar Huruf Braille Pada Cetakan 2 Sisi.....	90

Gambar 4.21 Hasil General Threshold dan Adaptive Threshold Pada Gambar Braille Cetakan 2 Sisi Gambar Dari Kamera Webcam.....	91
Gambar 4.22 Hasil Segmentasi Pada Cetakan 2 Sisi Gambar Dari Kamera.....	92
Gambar 4.23 Hasil Pengambilan Gambar Braille Cetakan 2 Sisi Dengan Alat Scanner	93
Gambar 4.24 Hasil General Threshold dan Adaptive Threshold Pada Gambar Braille Cetakan 2 Sisi Dari Scanner	94
Gambar 4.25 Hasil Segmentasi Cetakan 2 Sisi Gambar Dari Scanner	95

BAB 1

PENDAHULUAN

Seiring dengan perkembangan teknologi komputer, maka perkembangan serta aplikasi penggunaan teknologi komputer tersebut tumbuh secara pesat misalnya pada aplikasi pengolahan citra. Aplikasi pengolahan citra sudah banyak digunakan hampir semua bidang, salah satunya adalah bidang biomedika. Berdasarkan data dari badan kesehatan dunia (WHO) sekitar sepertiga populasi penyandang tuna netra di dunia tersebar di wilayah Asia Tenggara. Dari perkiraan ini, Indonesia dikatakan menjadi negara dengan jumlah penyandang tuna netra tertinggi yang populasinya mencapai 0,1-0,15% dari jumlah penduduk. Salah satu masalah yang dihadapi oleh penyandang cacat tuna netra adalah kesulitan dalam belajar membaca huruf braille. Oleh karena itu perlu dibutuhkan suatu sistem yang dapat membantu dalam proses belajar maupun membaca huruf braille, sehingga akan memudahkan tuna netra maupun orang umum dalam membaca huruf braille.

1.1 Latar Belakang

Salah satu cara untuk mendapatkan informasi adalah dengan membaca, namun bagi mereka yang mengalami kebutaan maka akan kesulitan jika harus membaca huruf biasa. Oleh karena itu didesain huruf khusus bagi orang yang mengalami gangguan penglihatan, yaitu huruf braille. Huruf braille tersusun dari enam titik timbul. Keenam titik tersebut dapat disusun sedemikian rupa sehingga menciptakan bermacam kombinasi. Cara untuk membaca cetakan huruf braille pada lembar kertas yaitu dengan menempelkan ujung jari-jari tangan pada lembar huruf braille dan meraba titik-titik timbul pada huruf braille tersebut.

Supaya bisa membaca huruf braille dibutuhkan waktu belajar yang cukup lama, perlu dilatih kepekaan tangan terhadap titik-titik timbul dan harus mengerti dan hafal kombinasi titik-titik timbul tersebut dalam membentuk suatu huruf. Penulis pernah berkunjung ke Sekolah Luar Biasa (SLB) yang ada di Lawang-Malang, dan bertemu dengan salah seorang pengajar yang bernama Bapak Jimmy.

Beliau mengatakan bahwa diperlukan waktu sekitar 2 tahun untuk belajar membaca huruf braille dengan baik. Sehingga banyak dari mereka yang mengalami tuna netra ternyata masih kesulitan dalam membaca huruf braille.

Untuk membantu dalam belajar dan membaca huruf *braille* telah dilakukan beberapa penelitian sebelumnya mengenai pengenalan huruf *braille* dengan sistem komputer melalui pengolahan citra. (Wajid dkk, 2011) telah melakukan penelitian pengenalan huruf braille dengan teknik pengolahan citra. Dalam penelitiannya digunakan alat scanner untuk mengambil gambar dari huruf braille, baru kemudian dilakukan pengolahan citra dan dihasilkan keluaran dalam bentuk teks. Penelitian lain telah dilakukan dengan mencoba pengambilan gambar dengan kamera handphone (Zhang dkk, 2007). Namun dalam percobaannya dilakukan pada huruf braille yang telah di pasang di tempat-tempat umum, bukan pada lembar kertas huruf braille.

Dari hasil pengamatan kami terhadap penelitian sebelumnya, cenderung menggunakan scanner untuk mengambil gambar huruf braille dan masih bersifat offline. Oleh karena itu dalam penelitian ini telah dibuat sistem pengenalan huruf braille yang bersifat real time. Pengambilan gambar huruf braille menggunakan kamera webcam, hasil dari kamera webcam akan secara langsung diproses dengan pengolahan citra, meliputi proses: grayscale, thresholding, erosion, dilation dan proses Blob Analysis. Baru kemudian digunakan metode artificial neural network untuk proses pengenalan huruf braille.

1.2 Perumusan Masalah

Pada penelitian ini terdapat permasalahan yang harus diselesaikan, yaitu :

1. Bagaimana menentukan pedoman pengambilan gambar huruf braille dengan kamera webcam agar didapat hasil gambar huruf braille yang baik.
2. Bagaimana melakukan proses pengolahan citra, supaya gambar hasil capture dari kamera dapat diproses dengan baik.
3. Bagaimana caranya membaca banyak titik timbul dalam gambar braille dan dapat membaca koordinat piksel x dan y tiap dot pada gambar braille.

4. Bagaimana caranya menentukan area segmentasi setiap satu huruf braille pada gambar braille.
5. Bagaimana membangun aplikasi artificial neural network untuk mengenali setiap huruf braille agar didapat hasil yang baik.

1.3 Batasan Masalah

Batasan masalah pada penelitian ini adalah:

1. Dalam pengambilan gambar braille dengan webcam, digunakan pencahayaan yang terang dan konstan.
2. Huruf braille sudah tercetak pada lembar kertas, dan kertas yang digunakan berwarna putih.
3. Cetakan huruf braille hanya pada satu sisi (single sheet).
4. Tempat meletakkan kertas braille yang akan dikenali sudah ditentukan.
5. Digunakan 2 webcam untuk pengambilan gambar braille.
6. Luas area cropping ditentukan secara manual terlebih dahulu.

1.4 Tujuan Dan Manfaat Penelitian

Tujuan dari penelitian ini adalah :

1. Mengetahui pedoman pengambilan gambar huruf braille dengan kamera webcam agar didapat hasil gambar huruf braille yang baik.
2. Mengetahui proses pengolahan citra, supaya gambar hasil capture dari kamera dapat diproses dengan baik.
3. Mengetahui caranya membaca banyak titik timbul dalam gambar braille dan dapat membaca koordinat piksel x dan y tiap dot pada gambar braille.
4. Mengetahui caranya menentukan area segmentasi setiap satu huruf braille pada gambar braille.
5. Mampu menerapkan metode artificial neural network untuk mengenali setiap huruf braille agar didapat hasil yang baik.

Manfaat dari penelitian ini adalah dapat dihasilkan suatu alat dan perangkat lunak (software) yang mampu mengenali pola-pola huruf braille, dan menterjemahkan mejadi teks. Sehingga dapat memudahkan tuna netra maupun orang awas dalam membaca tulisan huruf braille dan file teks yang dihasilkan dapat dijadikan file master dari buku, sehingga buku tersebut dapat dicetak ulang.

BAB 2

KAJIAN PUSTAKA DAN DASAR TEORI

Pada bab ini akan menjelaskan kajian pustaka dan dasar teori yang berhubungan dengan penelitian ini. Kajian pustaka berisi tentang penelitian sebelumnya yang berhubungan dengan penelitian ini. Sedangkan dasar teori berisi tentang teori-teori dasar yang akan digunakan dalam rencana penelitian ini.

2.1 Kajian Pustaka

Banyak teknik pengambilan citra huruf braille dan metode yang dikembangkan dalam penelitian untuk pengenalan huruf braille. Beberapa teknik pengambilan citra huruf braille antara lain pengambilan citra huruf braille menggunakan alat scanner dan kamera handphone. Sedangkan beberapa metode yang dikembangkan dalam pengenalan pola huruf braille pada penelitian sebelumnya antara lain metode Neural Network for Half Character, metode Cellular Neural Network (CNN), metode Support-Vector Machine Classifier dan metode Optical Braille Recognition (OBR).

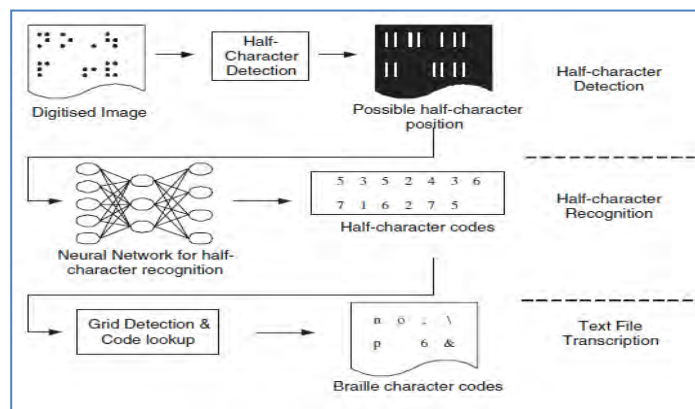
2.1.1 Pengenalan Huruf Braille Metode Neural Network for Half Character

Wong dkk (2004) melakukan penelitian melakukan pengenalan huruf braille dengan teknik pengolahan citra dan dengan menggunakan metode Neural Network for Half Character sehingga akan dihasilkan file teks. Gambar cetakan huruf braille pada kertas diambil terlebih dahulu menggunakan alat scanner CanonScan FB320P dengan resolusi 300dpi, contoh gambar hasil proses scanning huruf braille dengan scanner seperti terlihat pada Gambar 2.1.



Gambar 2.1 Hasil proses scanning huruf braille (Wong dkk, 2004)

Setelah dihasilkan gambar huruf braille, selanjutnya dilakukan proses pengolahan citra dan penerapan metode Neural Network for Half Character untuk pengenalan pola huruf braille. Algoritma deteksi Half Character menentukan keberadaan karakter dengan mendeteksi kemungkinan posisi dot pada gambar huruf braille. Deteksi dilakukan dengan cara setiap setengah karakter terlebih dahulu. Jika dalam satu karakter huruf braille terdiri atas dua kolom, maka dilakukan deteksi satu kolom dulu. Setelah setengah karakter telah diakui, kemudian akan ditentukan setengah karakter selanjutnya. Untuk mengetahui hasil dari setengah karakter yaitu kolom pertama, maka ditentukan setelah data input dimasukkan pada neural network, keluaran dari proses neural network berupa kode data. Sedangkan data setengah karakter selanjutnya yaitu kolom kedua, ditentukan dengan grid detection dan berdasarkan database yang sudah dibuat. Diagram sistem pada penelitian ini dapat diamati pada Gambar 2.2. Dari diagram sistem tersebut dapat dijelaskan pemrosesan gambar dimulai dari gambar sumber yang sudah berupa gambar biner, selanjutnya gambar biner tersebut dideteksi tiap area titik dot hitam dengan cara Half Character Detection, sehingga diketahui posisi setiap area dari titik dot. Kemudian dari data yang terbaca sebagai data input pada proses neural network untuk dihasilkan keluaran berupa kode, dari kode yang dihasilkan dicocokkan dengan data kode lookup table, dari hasil pencocokan dapat diketahui suatu huruf yang sesuai dengan kode. Sehingga dapat dihasilkan text file hasil penterjemahan.

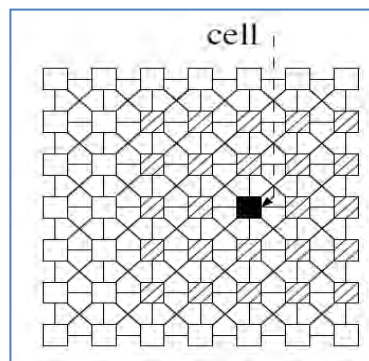


Gambar 2.2 Diagram alir sistem pengenalan huruf braille metode neural network for half-character (Wong dkk, 2004)

2.1.2 Pengenalan Huruf Braille Metode Cellular Neural Network

Metode lain yang di implementasikan dalam proses pengenalan huruf braille adalah metode Cellular Neural Network (CNN). Metode ini digunakan oleh Namba dkk (2006). Dalam penelitiannya untuk pengenalan huruf braille, gambar citra huruf braille diambil menggunakan kamera yang sudah tertanam pada handphone. Mereka mengatakan bahwa penggunaan kamera handphone cenderung lebih praktis dan bersifat real time dibandingkan dengan penggunaan alat scanner.

Dengan penggunaan metode CNN mereka memberi pernyataan bahwa metoda yang mereka gunakan mampu mencapai tingkat keberhasilan sebesar 87,9%. Teknik pada metode ini dengan memetakan lebih banyak area pada tiap satu karakter huruf braille. CNN mempunyai prinsip dasar seperti metode neural network pada umumnya. Yang membedakan adalah penentuan input yang akan dimasukkan ke layer input pada neural network tersebut. Pada metode CNN dalam menentukan input melalui pemetaan sel pada citra yang akan diproses. Objek citra yang akan di proses disegmentasi untuk mendapatkan pola-pola cell. Contoh pola cell terlihat pada Gambar 2.3. Pada gambar 2.3 terlihat pada cell utama hanya akan berhubungan dengan cell tetangganya saja. Dimana lebar cell tetangga tersebut harus ditentukan terlebih dahulu. Sebagai contoh pada Gambar 2.3 tersebut ditentukan bahwa jangkauan cell tetangga sebanyak 2 cell. Jika pada cell utama diketahui pada matriks [4,5] maka dapat di tentukan tabel matriks dari cell utama tersebut seperti terlihat pada Tabel 2.1.

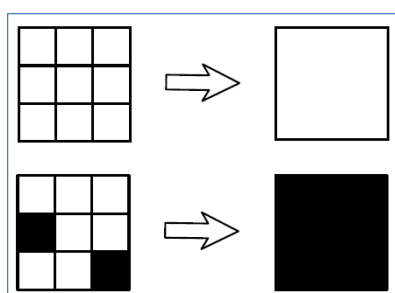


Gambar 2.3 Struktur cell pada CNN (Namba dkk, 2006)

TABEL 2.1 Matriks dari cell t_{45}

0	0	0	0	0	0	0
0	0	$t_{45}(-2,-2)$	$t_{45}(-2,-1)$	$t_{45}(-2,0)$	$t_{45}(-2,1)$	$t_{45}(-2,2)$
0	0	$t_{45}(-1,-2)$	$t_{45}(-1,-1)$	$t_{45}(-1,0)$	$t_{45}(-1,1)$	$t_{45}(-1,2)$
0	0	$t_{45}(0,-2)$	$t_{45}(0,-1)$	$t_{45}(0,0)$	$t_{45}(0,1)$	$t_{45}(0,2)$
0	0	$t_{45}(1,-2)$	$t_{45}(1,-1)$	$t_{45}(1,0)$	$t_{45}(1,1)$	$t_{45}(1,2)$
0	0	$t_{45}(2,-2)$	$t_{45}(2,-1)$	$t_{45}(2,0)$	$t_{45}(2,1)$	$t_{45}(2,2)$
0	0	0	0	0	0	0

Sumber: Namba dkk., 2006



Gambar 2.4 Penentuan nilai segmen cell pada CNN (Namba dkk, 2006)

Pada Gambar 2.4 dapat diamati cell yang bertetangga tersebut kemudian dikelompokkan tiap 9 segmen dimana nilai tiap segmen tersebut akan saling dikaitkan dan dihasilkan segmentasi yang lebih besar.

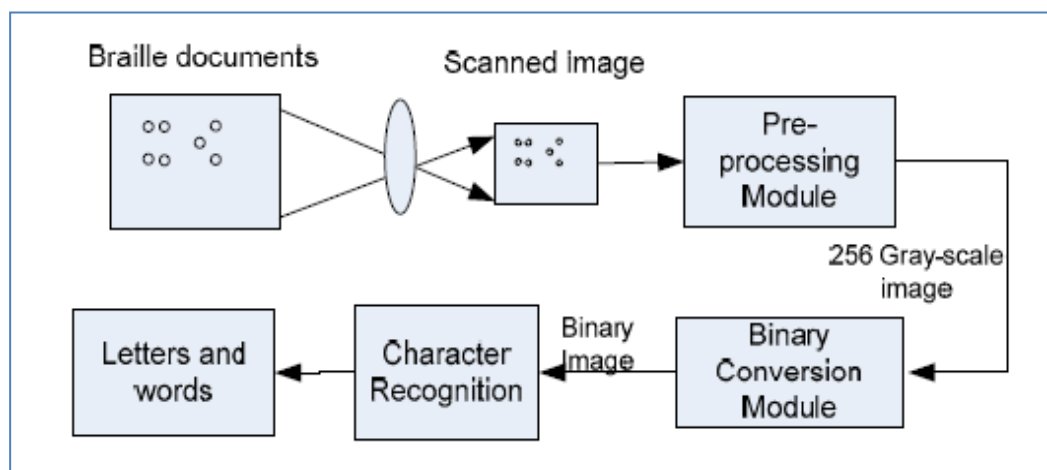
2.1.3 Pengenalan Huruf Braille Metode Support Vector Machine Classifier

Penelitian pengenalan huruf braille dengan teknik pengolahan citra dan dengan menggunakan metode Support-Vector Machine Classifier (SVM) untuk dihasilkan huruf dan teks (Li dkk., 2010). Pada penelitian ini Li dkk sudah menggunakan cetakan huruf braille yang tercetak dua sisi. Gambar cetakan huruf braille pada kertas diambil terlebih dahulu menggunakan alat scanner BenQ S6000 dengan resolusi 600 dpi, contoh gambar hasil proses scanning huruf braille dua dengan scanner seperti terlihat pada Gambar 2.5. Setelah dihasilkan gambar huruf braille, selanjutnya dilakukan proses persiapan pengolahan citra dan penerapan metode Support Vector Machine Classifier untuk pengenalan pola huruf braille untuk bisa diterjemahkan ke huruf teks.

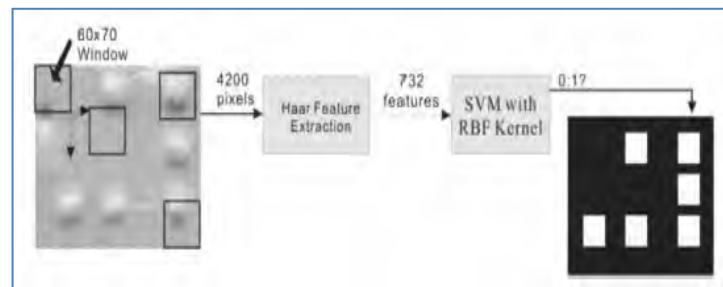


Gambar 2.5 Hasil proses scanning huruf braille cetakan dua sisi (Li dkk, 2010)

Persiapan pengolahan citra meliputi proses grayscale, proses grayscale adalah proses merubah citra warna RGB menjadi citra keabu-abuan (Li dkk., 2010). Pada penelitian ini sudah dilakukan cek kemiringan citra gambar huruf braille hasil dari alat scanner. Saat diketahui posisi miring maka dilakukan pelurusan terlebih dahulu. Setelah diketahui citra sudah lurus, kemudian dilakukan proses konversi citra ke citra biner. Citra biner merupakan citra yang memiliki nilai intensitas 0 dan 255, atau hitam dan putih saja. Proses menjadikan ke citra biner dilakukan agar data yang dihasilkan dari citra tersebut bisa dilanjutkan ke penerapan metode Support Vector Machine Classifier.



Gambar 2.6 Diagram sistem pengenalan huruf braille metode SVM Classifier (Li dkk., 2010)



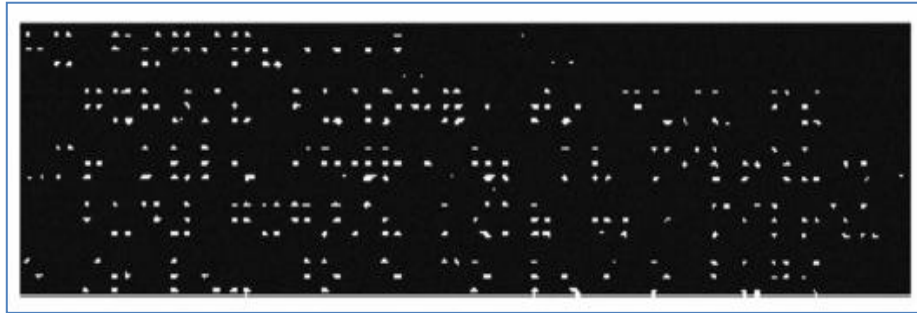
Gambar 2.7 Konversi citra grayscale menjadi citra biner menggunakan Haar Feature dan Support Vector Machine (Li dkk., 2010)

Pada Gambar 2.7 terlihat proses konversi dari citra grayscale dijadikan citra biner. Pada citra grayscale disegmentasi terlebih dahulu, setiap satu segmentasi terdiri atas 60x70 piksel, jadi total piksel tiap satu segmentasi berjumlah 4200 piksel. Kemudian data piksel tersebut diproses dengan Haar Feature Extraction sehingga menghasilkan 732 fitur, selanjutnya 732 fitur tersebut diproses dengan SVM untuk diketahui pada segmen tersebut merupakan titik dot apa tidak.

Pada penelitian ini digunakan tiga jenis wavelet haar untuk menghitung fitur dari vektor, seperti terlihat pada Gambar 2.8. Pertama digunakan untuk mendeteksi tepi arah vertikal, kedua untuk tepi arah horisontal dan ketiga untuk tepi diagonal (Li dkk, 2010). Pada metode SVM ini diklarifikasikan bahwa pada gambar huruf braille yang mengandung titik dot memiliki data sampel positif dan gambar huruf braille yang tidak mengandung titik dot memiliki data sampel negatif. Dengan menggunakan metode SVM ini maka akan dapat ditemukan batas jarak untuk memisahkan data sampel positif dan data sampel negatif (Li dkk, 2010).



Gambar 2.8 Tiga jenis tipe wavelet haar pada metode SVM (Li dkk, 2010)

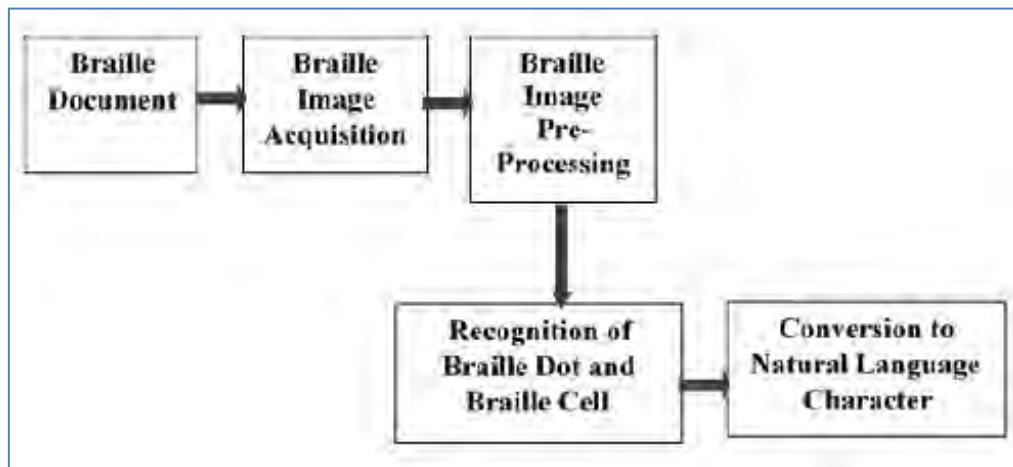


Gambar 2.9 Hasil proses citra biner dengan SVM (Li dkk., 2010)

Pada Gambar 2.9 merupakan hasil keluaran dari proses citra biner dengan SVM. Dengan sudah dideteksi citra dari titik dot maka proses selanjutnya maka didapat nilai data tiap segmentasi satu karakter huruf braille, data tersebut selanjutnya dimasukkan ke proses metode SVM, dimana SVM yang sebelumnya sudah dilatih. Sehingga dihasilkan suatu data keluaran berupa kode data, dan kode data tersebut dapat dikonversi ke huruf teks.

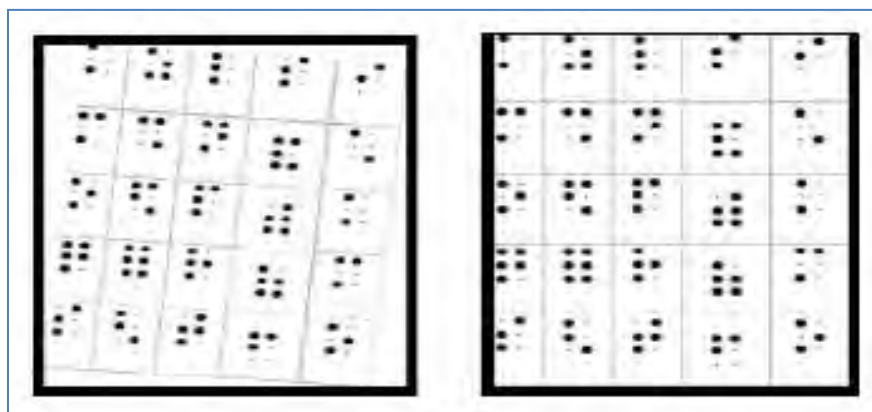
2.1.4 Pengenalan Huruf Braille Metode Optical Braille Recognition

Metode lain yang digunakan adalah Optical Braille Recognition (OBR), merupakan metode dengan memetakan setiap segmen dot pada titik-titik huruf braille. Dalam cara segmentasi yang dilakukan area tiap satu karakter dibagi lagi menjadi 6 segmen, kemudian membaca nilai piksel tiap segmen sehingga dapat ditentukan nilai tiap segmen tersebut (Shreekanth dkk., 2013). Pada penelitian ini dilakukan proses pengenalan pola huruf braille pada cetakan lembar kertas. Langkah-langkah yang dilakukan meliputi: meng-akuisisi gambar, koreksi pada citra miring, pemrosesan gambar, men-segmentasi gambar braille, dan terakhir pencocokan pola yang dibaca dengan pola database yang sudah disimpan. Pada proses akuisisi gambar lembar kertas cetakan huruf braille discan menggunakan alat scanner, sehingga akan menghasilkan gambar yang akan diproses, dari beberapa hasil yang didapat dengan seting piksel 80 dpi sampai 200 dpi. Pada Gambar 2.10 merupakan diagram alir sistem yang digunakan dalam penelitian ini.

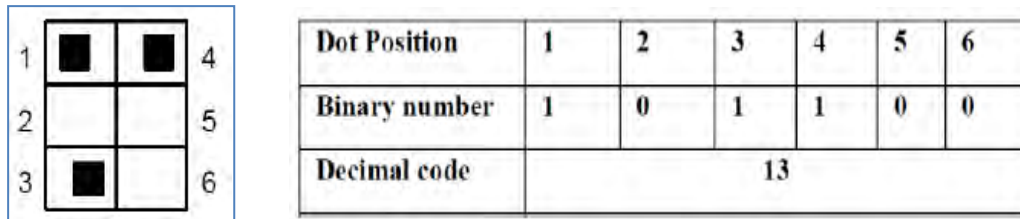


Gambar 2.10 Diagram sistem pengenalan huruf braille metode OBR
(Shreekanth dkk, 2013)

Proses selanjutnya dilakukan koreksi posisi gambar yang diproses, tegak atau miring. Jika tegak maka tidak ada masalah dan dilanjutkan proses selanjutnya, namun jika posisinya miring maka dilakukan perbaikan dengan memutar piksel-piksel pada gambar. Proses selanjutnya adalah proses menghilangkan noise pada gambar, noise disini diasumsikan sebagai gambar piksel yang muncul, dimana kemunculannya tersebut seharusnya tidak sesuai pola dari huruf braille. Dalam proses perbaikan ini dilakukan teknik: korelasi silang, deteksi tepi dan thresholding.



Gambar 2.11 (a) Diketahui posisi gambar miring, (b) Setelah dilakukan proses pelurusan posisi gambar (Shreekanth dkk, 2013)

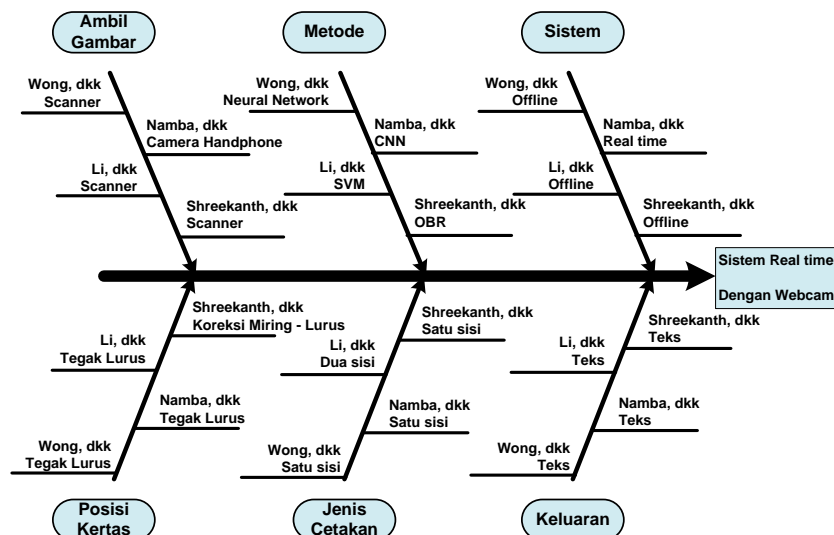


Gambar 2.12 (a) Hasil proses segmentasi, (b) Proses perhitungan nilai data dari hasil segmentasi (Shreekanth dkk., 2013)

Pada Gambar 2.12 menunjukkan proses segmentasi dot tiap pola huruf braille, pada proses ini menggunakan Laplacian operator Gaussian G ordinaly yang diterapkan pada seluruh gambar untuk mengidentifikasi tepi piksel . proses terakhir adalah mengidentifikasi pola yang dihasilkan untuk dicocokkan dengan pola database huruf braille yang sudah ditentukan, sehingga akan di dapat nilai data. Sehingga dapat ditentukan setiap karakter huruf braille tersebut kedalam karakter alphabet berdasarkan nilai data yang dihasilkan.

2.1.5 Fishbone Diagram

Berdasarkan kajian pustaka pada penelitian sebelumnya maka dibuat fishbone diagram seperti terlihat pada Gambar 2.13 berikut:



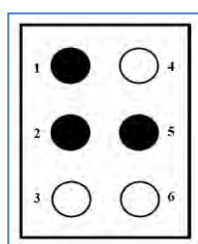
Gambar 2.13 Fishbone diagram penelitian sebelumnya

2.2 Dasar Teori

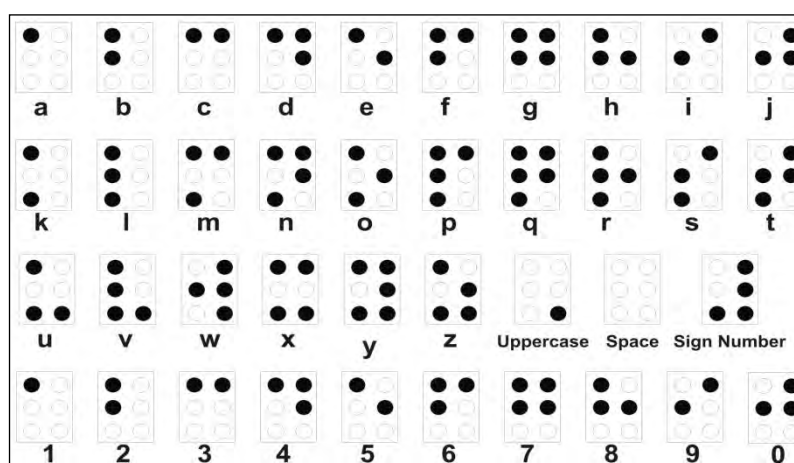
Pada sub bab ini akan menjelaskan beberapa dasar teori yang berkaitan dengan penelitian yang dilakukan. Berikut adalah dasar teori yang berkaitan dengan penelitian ini:

2.2.1 Huruf Braille

Huruf Braille adalah suatu sistem penulisan yang menggunakan titik-titik yang timbul yang mewakili karakter tertentu. Huruf yang diciptakan oleh Louis Braille ini diperuntukkan untuk orang tuna netra. Satuan dasar dari sistem tulisan ini disebut sel braille, di mana tiap sel terdiri dari enam titik timbul, tiga baris dengan dua titik. Keenam titik tersebut dapat disusun sedemikian rupa hingga menciptakan bermacam kombinasi. Pada Gambar 2.14 ditunjukkan adanya warna hitam dan putih, pada gambar tersebut dapat diasumsikan bahwa lingkaran yang berwarna hitam menunjukkan dot yang timbul sedangkan yang putih merupakan dot yang tidak timbul.



Gambar 2.14 Susunan titik-titik dot pada huruf braille



Gambar 2.15 Susunan kombinasi titik-titik timbul pada huruf braille



Gambar 2.16 Membaca huruf braille dengan diraba

Pada Gambar 2.15 menunjukkan susunan kombinasi titik-titik pada huruf braille dalam mewakili huruf abjad. Setiap karakter memiliki kombinasi yang berbeda, sehingga dengan 6 titik tersebut maka bisa dibuat sebanyak 64 kombinasi. Sedangkan pada Gambar 2.16 ditunjukkan bagaimana cara membaca huruf braille pada umumnya yang dilakukan oleh tuna netra, yaitu dengan menempelkan ujung jari-jari tangan pada lembar huruf braille dan meraba titik-titik timbul pada huruf braille. Perlu dilatih kepekaan tangan dalam membaca huruf braille tersebut dan harus hafal setiap kombinasi titik-titik yang terbentuk. Sehingga proses untuk dapat membaca huruf braille dibutuhkan waktu yang relatif lama.

2.2.2 Pengolahan Citra

Pengolahan citra merupakan teknik manipulasi citra secara digital yang khususnya menggunakan komputer untuk digunakan dalam suatu aplikasi tertentu. Pengolahan citra pada umumnya dilakukan dengan tujuan untuk memperbaiki kualitas suatu gambar sehingga dapat dengan mudah oleh pengguna untuk mengolah informasi yang ada pada suatu gambar untuk kebutuhan identifikasi. Secara objektif tujuan dari pengolahan citra adalah mengubah dan menganalisis suatu informasi berupa gambar sehingga informasi baru tentang gambar tersebut dapat dibuat lebih jelas yang selanjutnya dihubungkan dengan kemampuan otak manusia.

Dalam perkembangan pengolahan citra digunakan sebagai proses klasifikasi dalam kebutuhan identifikasi, ada empat klasifikasi dasar yaitu:

- a. Point digunakan untuk memproses nilai piksel suatu gambar berdasarkan posisi piksel tersebut.
- b. Area digunakan untuk memproses nilai berdasarkan kumpulan piksel.
- c. Geometric digunakan untuk mengubah posisi suatu piksel.
- d. Frame digunakan untuk memproses berdasarkan 2 buah gambar atau lebih.

Citra digital dapat berupa suatu gambar yang terbentuk dari susunan piksel dalam baris dan kolom, dimana setiap titiknya memiliki informasi tertentu. Pada sebuah gambar berwarna pada umumnya menggunakan model warna RGB (Red-Green-Blue), dimana pada masing-masing titik piksel memiliki nilai informasi tingkat kecerahan warna primer yang memiliki batasan sebesar 8 bit pada setiap komponen warna. Maka total untuk satu piksel warna direpresentasikan dengan nilai 24 bit.

A. Grayscale

Citra berskala keabuan (grayscale) adalah citra yang menggunakan gradasi warna abu-abu yang merupakan kombinasi antara hitam dan putih (Kadir, 2013). Setiap warna di dalam citra berskala keabuan dinyatakan dengan sebuah nilai bulat antara 0 sampai 255, dan nilai tersebut disebut nilai intensitas.

Proses pengubahan gambar dari format RGB menjadi format grayscale dilakukan dengan menggunakan metode illuminance grayscale yang direpresentasikan dalam Persamaan 2.1 berikut:

$$\text{Grayscale} = 0,299R + 0,587G + 0,114B \quad (2.1)$$

Dimana R adalah nilai intensitas warna Red pada piksel gambar, G adalah nilai intensitas warna Green pada piksel gambar, B adalah nilai intensitas warna Blue pada piksel gambar. Grayscale adalah nilai intensitas pada piksel gambar (keabuan).

B. Thresholding

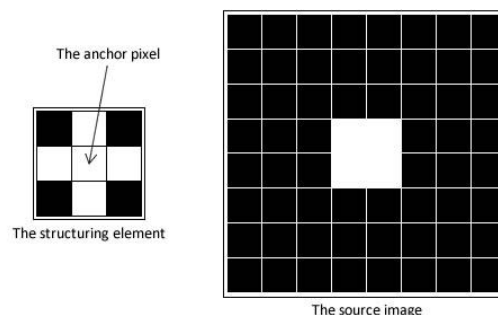
Proses thresholding atau binarisasi pada prinsipnya adalah melakukan pengubahan nilai intensitas warna piksel menjadi 2 nilai yaitu 0 atau warna hitam dan 255 atau warna putih. Pengubahan diharapkan akan menghasilkan citra yang hanya terdiri dari dua warna saja, sehingga untuk proses selanjutnya menjadi lebih sederhana. Jenis proses threshold yang digunakan diimplementasikan dengan Persamaan 2.2.

$$g(x, y) = \begin{cases} 255 & \text{jika } f(x, y) \geq Th \\ 0 & \text{untuk yang lain} \end{cases} \quad (2.2)$$

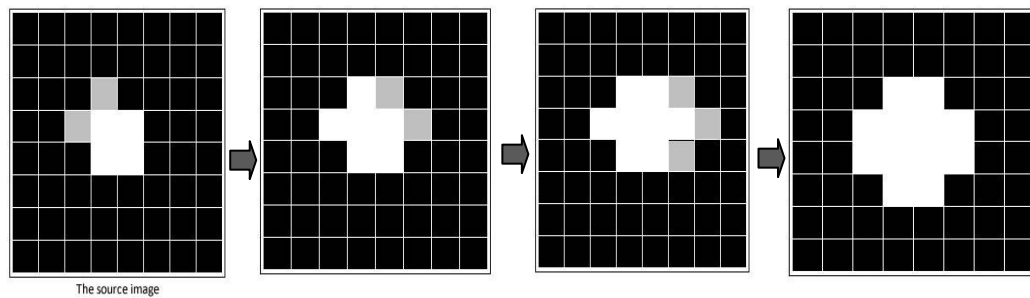
Dimana Th adalah nilai threshold yang ditentukan. Pada proses ini menghasilkan suatu citra dengan warna hitam dan putih yang akan digunakan untuk proses lebih lanjut. Tujuan utama dari proses thresholding ini adalah untuk memisahkan dan membedakan antara objek dengan latar belakang (background) dari suatu citra. Sehingga dapat diketahui nilai tiap piksel dari citra tersebut bernilai 0 atau bernilai 255.

C. Dilatasi

Dilatasi merupakan proses menambahkan atau mempertebal objek pada gambar biner. Hal ini dilakukan dengan merubah nilai pixel pada batasan suatu objek pada suatu gambar. Jumlah pixel yang dilakukan perubahan tergantung dari ukuran dan bentuk dari structuring element yang digunakan pada pengolahan gambar.



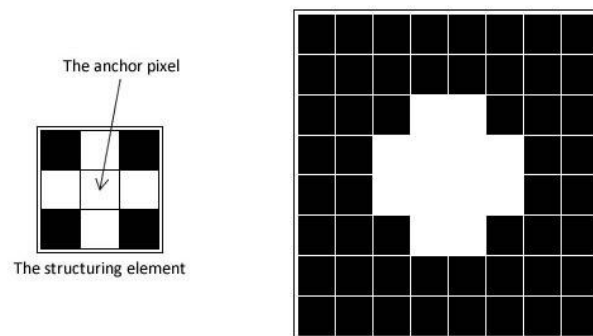
Gambar 2.17 Structure element dan gambar yang akan diproses dilatasi



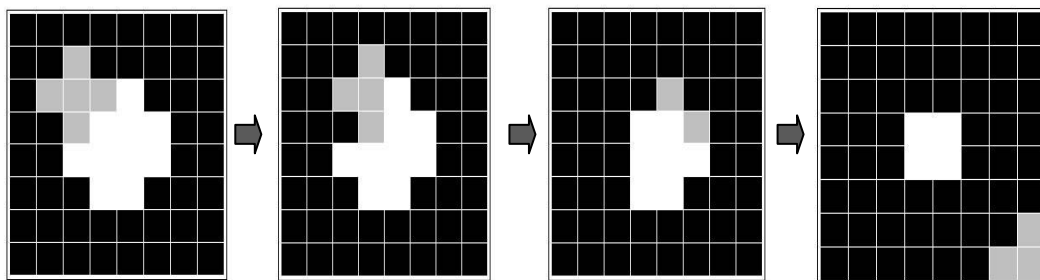
Gambar 2.18 Ilustrasi proses dilatasi

D. Erosi

Erosi merupakan proses untuk mengecilkan ataupun menipiskan objek pada gambar. Hal ini dilakukan dengan merubah nilai pixel pada batasan suatu objek pada suatu gambar. Sama seperti dilatasi, jumlah pixel yang dilakukan perubahan tergantung dari ukuran dan bentuk dari structuring element yang digunakan pada pengolahan gambar.



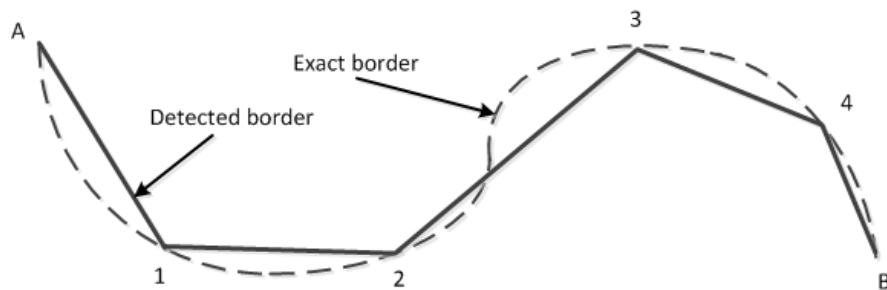
Gambar 2.19 Structure element dan gambar yang akan diproses erosi



Gambar 2.20 Ilustrasi proses erosi

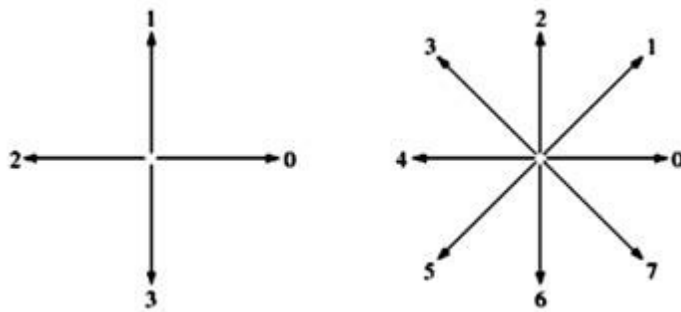
E. Blob Analysis

Proses blob analysis dilakukan untuk menentukan keberadaan sebuah objek yang berada di dalam suatu kontur tertutup pada gambar citra digital (Sonka, dkk., 2008). Di bidang pemrosesan citra, nilai kontur menggambarkan sebuah nilai sequence (urutan). Kontur dapat ditemukan jika ada titik piksel yang mempunyai lokasi tepi yang signifikan pada gambar, maka dapat dilakukan pendekatan tepi yang sesuai dengan arah tepi yang diasumsikan. Proses iterasi yang berturut-turut pada tepi yang berupa titik-titik, dan menghubungkannya dengan garis lurus sehingga membentuk sebuah prediksi garis tepi seperti yang ditunjukkan pada Gambar 2.21.



Gambar 2.21 Hubungan garis lurus membentuk prediksi garis tepi
(Sonka dkk, 2008)

Salah satu pendekatan untuk mendeteksi kontur yang telah diperkenalkan adalah chain code (kode rantai) pertama kali diperkenalkan untuk merepresentasikan kurva digital oleh Herbert Freeman, karena itu chain code disebut juga dengan Freeman code (Sonka, dkk., 2008). Menurut Freeman, skema coding untuk struktur garis harus memenuhi 3 syarat, yaitu menjaga informasi penting tidak hilang, dapat disimpan dan ditampilkan lagi dan mudah dilakukan pengolahan yang diperlukan. Chain code digunakan untuk menggambarkan batas obyek atau jumlah piksel yang berada dalam satu obyek. Batas obyek direpresentasikan dengan piksel-piksel yang saling terhubung dan memiliki nilai yang sama. Kode rantai berjalan dengan menelusuri piksel-piksel pada citra berdasarkan prioritas arah yang telah ditentukan. Sebuah kode rantai bisa terdiri dari 4 arah mata angin atau 8 arah mata angin seperti pada Gambar 2.22.



Gambar 2.22 Kode rantai arah 4 mata angin dan 8 mata angin
(Sonka, dkk., 2008)

Arah dari tiap segmen direpresentasikan dengan angka tertentu. Elemen pertama pada sebuah urutan harus memberikan informasi mengenai posisinya sehingga rekonstruksi area atau perhitungan luas dapat dilakukan. Dimulai dari sebuah pixel tepi dan searah jarum jam, arah setiap pixel tepi yang membentuk batas objek dikodekan dengan salah satu dari delapan kode rantai. Kode rantai merepresentasikan batas objek dengan koordinat pixel tepi pertama lalu diikuti dengan senarai kode rantai.

F. Segmentasi Area

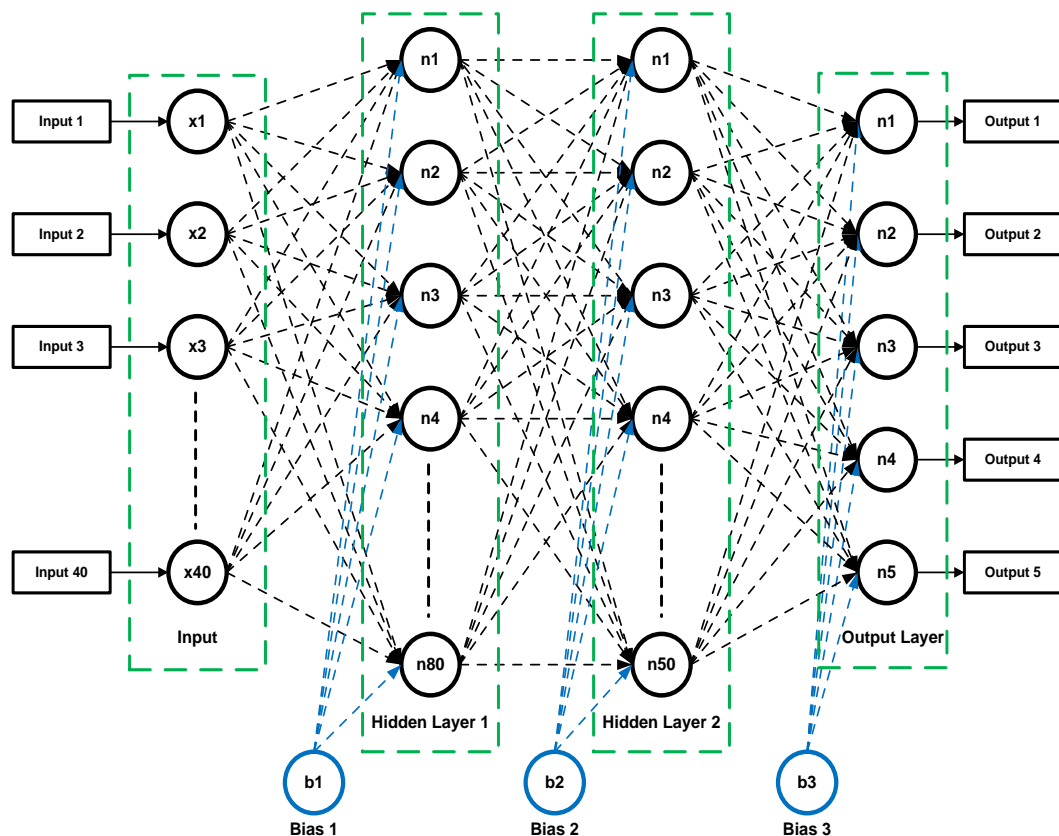
Segmentasi adalah suatu proses pemisahan objek yang satu dengan objek yang lain dalam suatu gambar citra digital. Pemisahan objek dapat dilakukan menggunakan suatu pendekatan daerah dengan mengidentifikasi nilai tertentu pada masing-masing titik piksel daerah tersebut. Segmentasi memiliki tiga macam cara pendekatannya, antara lain:

- a. Region approach, pendekatan daerah digunakan untuk memisahkan beberapa daerah untuk mendapatkan daerah sesuai yang diinginkan.
- b. Edge approach, pendekatan tepi untuk mendapatkan titik-titik tepi yang saling terhubung mengelilingi suatu daerah.
- c. Boundary approach, pendekatan batas digunakan untuk mendapatkan batas antar daerah.

Hasil dari proses segmentasi pada citra digital akan membentuk suatu kumpulan piksel menjadi luasan daerah yang memiliki suatu nilai tertentu sesuai yang diinginkan.

2.2.3 Artificial Neural Network

Artificial neural network atau bisa disebut jaringan syaraf tiruan merupakan salah satu representasi buatan dari otak manusia yang selalu mencoba untuk mensimulasikan proses pembelajaran pada otak manusia tersebut. Istilah buatan disini digunakan karena jaringan syaraf ini diimplementasikan dengan menggunakan program komputer yang mampu menyelesaikan sejumlah proses perhitungan selama proses pembelajaran (Kusumadewi dkk, 2006). Pada Gambar 2.23 contoh dari topologi dari Artificial Neural Network (ANN).



Gambar 2.23 Arsitektur topologi artificial neural network.

Seperti halnya otak manusia, jaringan syaraf juga terdiri dari beberapa neuron, dan ada hubungan antara neuron-neuron tersebut. Neuron-neuron tersebut akan mentransformasikan informasi yang diterima melalui sambungan keluarnya menuju ke neuron-neuron yang lain. Pada jaringan syaraf penghubung antar neuron tersebut dikenal dengan nama bobot. Nilai bobot itu yang nantinya menentukan hasil keluaran dari proses ANN. Dimana nilai bobot optimal dihasilkan dari proses pembelajaran terlebih dahulu. Jadi sebelum ANN dijalankan untuk proses suatu data, maka pada ANN tersebut dilakukan proses pembelajaran terlebih dahulu, proses ini bertujuan untuk mendapatkan nilai bobot yang optimal pada tiap-tiap layer. Salah satu metode pembelajaran yang digunakan dalam ANN adalah Backpropagation. Backpropagation memiliki arsitektur jaringan yang tidak memiliki koneksi umpan balik tetapi error dipropagasikan selama proses pembelajaran. Pada pembelajaran ini digunakan Error dengan perhitungan Mean Square Error (MSE).

A. Karakteristik Neural network

Penyelesaian masalah dengan neural network tidak memerlukan pemrograman. Neural network menyelesaikan masalah melalui proses belajar dari contoh-contoh pelatihan yang diberikan. Biasanya pada neural network diberikan sebuah himpunan pola pelatihan yang terdiri dari sekumpulan contoh pola. Proses belajar neural network berasal dari serangkaian contoh-contoh pola yang diberikan. Metode pelatihan yang sering dipakai adalah metode belajar terbimbing. Selama proses belajar itu pola masukan disajikan bersama-sama dengan pola keluaran yang diinginkan. Jaringan akan menyesuaikan nilai bobotnya sebagai tanggapan atas pola masukan dan sasaran yang disajikan tersebut.

1. Faktor Bobot

Bobot merupakan suatu nilai yang mendefinisikan tingkat atau kepentingan hubungan antara suatu node dengan node yang lain. Semakin besar bobot suatu hubungan menandakan semakin pentingnya hubungan kedua node tersebut. Bobot merupakan suatu hubungan berupa bilangan real maupun integer, tergantung dari

jenis permasalahan dan model yang digunakan. Bobot-bobot tersebut bisa ditentukan untuk berada didalam interval tertentu. selama proses pelatihan, bobot tersebut dapat menyesuaikan dengan pola-pola input.

Jaringan dengan sendirinya akan memperbaiki diri terus-menerus karena adanya kemampuan untuk belajar. Setiap ada suatu masalah baru, jaringan dapat belajar dari masalah baru tersebut, yaitu dengan mengatur kembali nilai bobot untuk menyesuaikan karakter nilai (Kusumadewi dkk., 2006).

2. Fungsi Aktivasi

Setiap neuron mempunyai keadaan internal yang disebut level aktivasi atau level aktivitas yang merupakan fungsi input yang diterima. Secara tipikal suatu neuron mengirimkan aktivitasnya kebeberapa neuron lain sebagai sinyal.

Ada beberapa pilihan fungsi aktivasi yang digunakan dalam metode backpropagation, seperti fungsi sigmoid biner, dan sigmoid bipolar. Karakteristik yang harus dimiliki fungsi aktivasi tersebut adalah kontinyu, diferensiabel, dan tidak menurun secara monoton. Berikut ini adalah fungsi aktivasi yang sering digunakan yaitu:

2.A Fungsi Sigmoid Biner

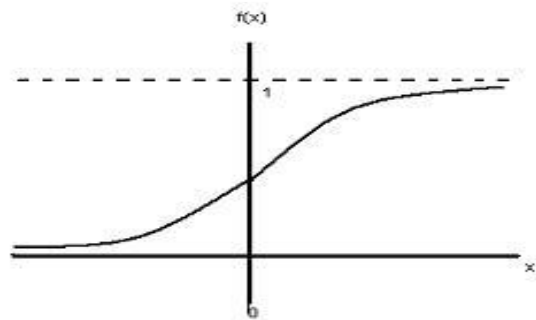
Fungsi ini digunakan untuk neural network yang dilatih dengan menggunakan metode backpropagation. Fungsi sigmoid biner memiliki nilai pada range 0 sampai 1. Fungsi ini sering digunakan untuk neural network yang membutuhkan nilai output yang terletak pada interval 0 sampai 1. Definisi fungsi sigmoid biner adalah sebagai berikut:

$$f_1(x) = \frac{1}{(1 + e^{-x})} \quad (2.3)$$

dengan turunan :

$$f'_1(x) = f_1(x)(1 - f_1(x)) \quad (2.4)$$

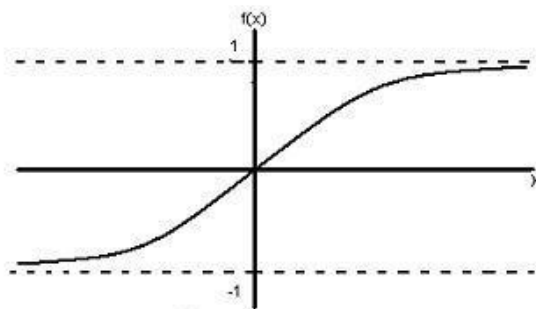
berikut ini fungsi sigmoid biner:



Gambar 2.24 Fungsi sigmoid biner

2.B Fungsi Sigmoid Bipolar

Fungsi sigmoid bipolar hampir sama dengan fungsi sigmoid biner, hanya saja output dari fungsi ini memiliki range antara 1 sampai -1. Gambar 2.25 menunjukkan area dari fungsi sigmoid bipolar.



Gambar 2.25 Fungsi sigmoid bipolar.

Definisi fungsi sigmoid bipolar adalah sebagai berikut :

$$f_2(x) = 2 f_1(x) - 1 \quad (2.5)$$

dengan turunan :

$$f_2'(x) = \frac{1}{2} (1 + f_2(x))(1 - f_2(x)) \quad (2.6)$$

B. Model Neural Network Backpropagation

Pelatihan pada neural network backpropagation, feedfoward (umpan maju) dilakukan dalam rangka perhitungan bobot sehingga pada akhir pelatihan akan diperoleh bobot-bobot yang baik. Selama proses pelatihan, bobot-bobot diatur secara iteratif untuk meminimalkan error (kesalahan) yang terjadi. Error (kesalahan) dihitung berdasarkan rata-rata kuadrat kesalahan (MSE). Rata-rata kuadrat kesalahan juga dijadikan dasar perhitungan untuk kerja fungsi aktivasi. Sebagian besar pelatihan untuk jaringan feedfoward (umpan maju) menggunakan gradien dari fungsi aktivasi untuk menentukan bagaimana mengatur bobot-bobot dalam rangka meminimumkan kinerja. Gradien ini ditentukan dengan menggunakan suatu teknik yang disebut backpropagation.

Algoritma pelatihan standar backpropagation akan menggerakkan bobot dengan arah gradien negatif. Prinsip dasar dari algoritma backpropagation adalah memperbaiki bobot-bobot jaringan dengan arah yang membuat fungsi aktivasi menjadi turun dengan cepat. Pelatihan backpropagation meliputi tiga tahapan sebagai berikut :

B.1 Propagasi Maju

Pola masukan dihitung maju mulai dari input layer hingga output layer menggunakan fungsi aktivasi yang ditentukan.

Tahap 1 :

Masing-masing unit hidden layer dikalikan weight dan dijumlahkan dengan threshold :

$$Z_{in_j} = V_{o_j} + \sum_{i=1}^n X_i V_{ij} \quad (2.7)$$

Dihitung dengan fungsi aktifasi :

$$Z_j = f(Z_{in_j}) \quad (2.8)$$

Kemudian dihitung pada fungsi sigmoid yang bentuk fungsinya adalah :

$$Z_j = \frac{1}{1 + \exp^{(-z_{in_j})}} \quad (2.9)$$

Sinyal keluaran dari fungsi aktivasi tersebut dikirimkan ke semua unit di output layer.

Tahap 2 :

Masing-masing output layer dikalikan dengan weight serta ditambah dengan threshold :

$$Y_{in_k} = W_{0_k} + \sum_{j=1}^p Z_j W_{jk} \quad (2.10)$$

Kemudian dihitung kembali sesuai dengan fungsi aktivasi :

$$y_k = f(y_{in_k}) \quad (2.11)$$

B.2 Propagasi Mundur

Selisih antara keluaran jaringan dengan target yang diinginkan merupakan kesalahan yang terjadi. Kesalahan yang terjadi itu dipropagasi mundur.

Tahap 1 :

Masing-masing output layer menerima pola target sesuai dengan pola masukan saat training dan dihitung errornya :

$$\delta_k = (t_k - y_k)f'(y_{in_k}) \quad (2.12)$$

Karena $f(y_{in_k}) = y_k$ menggunakan fungsi sigmoid, maka :

$$f'(y_{in_k}) = f(y_{in_k})(1 - f(y_{in_k})) \quad (2.13)$$

Menghitung perbaikan weight, kemudian digunakan untuk memperbaiki W_{jk} :

$$\Delta W_{jk} = \alpha \cdot \partial_k \cdot Z_j \quad (2.14)$$

Menghitung perbaikan koreksi Dan menggunakan nilai delta (d_k) pada semua layer sebelumnya.

$$\Delta W_{0k} = \alpha \cdot \partial_k \quad (2.15)$$

Tahap 2 :

Masing-masing weight yang menghubungkan output layer dengan hidden layer dikalikan delta (d_k) dan dijumlahkan sebagai masukan untuk layer berikutnya :

$$\delta_{in_j} = \sum_{k=1}^m \delta_k W_{jk} \quad (2.16)$$

Selanjutnya dikalikan dengan turunan fungsi aktivasi untuk menghitung error.

$$\delta_j = \delta_{in_j} f'(y_{in_j}) \quad (2.17)$$

Langkah berikutnya menghitung perbaikan weight :

$$\Delta V_{ij} = \alpha \delta_j X_i \quad (2.18)$$

Kemudian menghitung perbaikan threshold :

$$\Delta V_{0j} = \alpha \delta_j \quad (2.19)$$

B.3 Perubahan Bobot

Modifikasi bobot untuk menurunkan kesalahan yang terjadi. Ketiga fase tersebut diulang-ulang terus hingga kondisi penghentian dipenuhi.

Tahap 1 :

Masing-masing outputlayer diperbaiki threshold dan weight :

$$W_{jk}(\text{baru}) = W_{jk}(\text{lama}) + \Delta W_{jk} \quad (2.20)$$

Tahap 2 :

Masing-masing hidden layer diperbaiki threshold dan weight :

$$V_{jk}(\text{baru}) = V_{jk}(\text{lama}) + \Delta V_{jk} \quad (2.21)$$

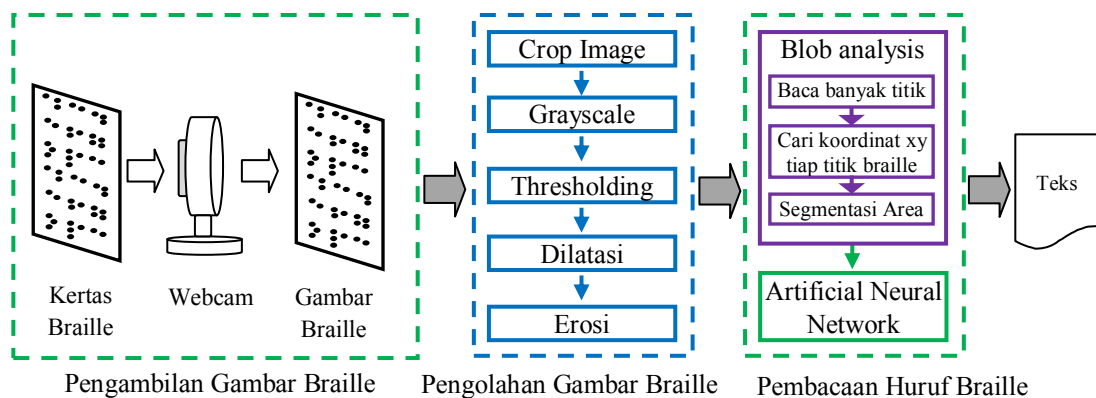
Keterangan :

- X_i : nilai aktivasi dari unit X_i
- Z_j : unit ke-j pada hidden layer
- Z_{in_j} : output untuk unit Z_j
- Y_k : unit ke-k pada outputlayer
- Y_{in_k} : output untuk unit Y_k
- y_k : nilai aktivasi dari unit Y_k
- W_{ok} : nilai weight pada threshold untuk unit Y_k
- W_{kj} : nilai weight dari Z_j ke unit Y_k
- ΔW_{kj} : selisih antara $W_{kj}(t)$ dengan $W_{kj}(t+1)$
- V_{jo} : nilai weight pada threshold untuk unit Z_j
- V_{ij} : nilai weight dari unit X_i ke unit Z_j
- ΔV_{ij} : selisih antara $V_{ij}(t)$ dengan $V_{ij}(t+1)$
- δ_k : faktor pengendalian nilai weight pada outputlayer
- δ_j : faktor pengendalian nilai weight pada hidden layer
- α : konstanta laju training.

BAB 3

METODE PENELITIAN

Penelitian ini dikhususkan pada proses pengenalan huruf braille secara real time. Untuk pengambilan gambar huruf braille digunakan webcam kamera, dari gambar yang dihasilkan webcam selanjutnya dilakukan proses pengolahan citra meliputi: proses cropping, proses grayscale, proses thresholding, proses dilation dan proses erosion. Dari hasil pengolahan citra tersebut, kemudian dilakukan pencarian dan pembacaan letak koordinat gambar titik hitam pada gambar braille dengan menggunakan teknik blob analysis. Dengan teknik blob analysis dapat dideteksi berapa banyak titik hitam pada area gambar braille serta dapat diketahui juga posisi koordinat x dan y tiap titik hitam tersebut. Data koordinat x dan y yang telah diketahui akan dijadikan acuan untuk menentukan area segmentasi pada setiap satu huruf braille. Hasil segmentasi area pada satu huruf braille disegmentasi lagi menjadi beberapa area segmentasi kecil lagi. Nilai piksel dari area segmentasi kecil tersebut akan dijadikan nilai input pada proses pengenalan huruf braille melalui metode artificial neural network. Proses pembacaan tiap satu huruf braille dengan artificial neural network akan didapatkan satu huruf teks. Pembacaan dilakukan terus sampai semua area gambar braille terbaca. Diagram blok dari sistem secara keseluruhan adalah seperti ditunjukkan pada Gambar 3.1.



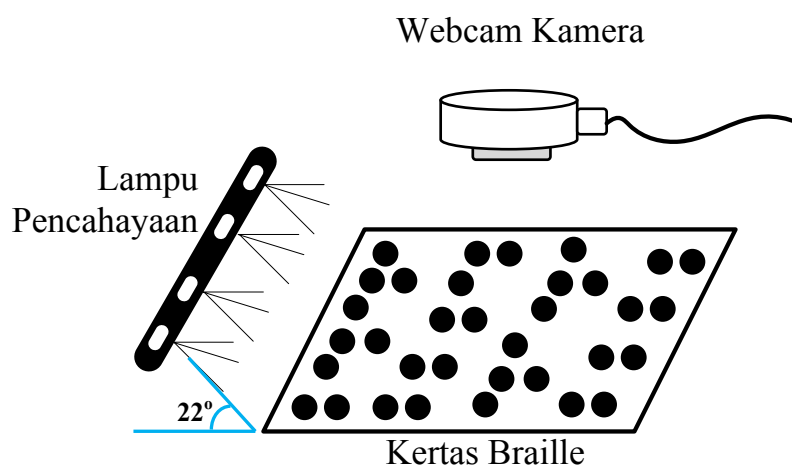
Gambar 3.1 Diagram blok sistem proses pembacaan huruf braille

Berdasarkan Gambar 3.1, untuk mempermudah dalam proses perencanaan dan pembuatan, sistem dibagi menjadi tiga bagian yaitu :

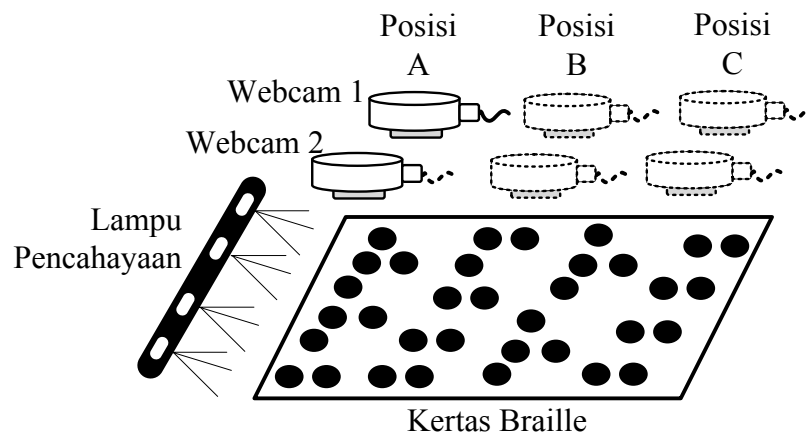
1. Pengambilan gambar braille
2. Pengolahan gambar braille
3. Pembacaan huruf braille

3.1 Pengambilan Gambar Braille

Pengambilan gambar braille dilakukan dengan menggunakan webcam kamera dan dihubungkan dengan komputer atau laptop. Webcam kamera yang digunakan memiliki resolusi VGA (640 x 480). Teknik pengambilan gambar braille agar di dapat hasil yang baik, maka pada objek yaitu lembar kertas braille diberikan penerangan yang cukup. Penempatan sumber pencahayaan pada lembar kertas braille tidak di letakkan pada sisi atas kertas braille langsung, namun sumber pencahayaan dilakukan dari sisi samping kertas braille dengan sudut kemiringan pencahayaan sekitar 22° terhadap kertas braille. Pencahayaan diberikan dari sisi samping karena akan membentuk bayangan pada tiap titik timbul pada cetakan huruf braille, sehingga gambar titik timbul huruf braille hasil capture dari webcam kamera akan terlihat jelas. Kertas lembar braille pada umumnya berukuran lebar 24,8cm dan panjang 30,5cm. Dengan ukuran tersebut pada lembar kertas braille biasanya cetakan huruf braille terdiri atas 26 baris dan tiap baris menampung 35 karakter huruf braille. Ilustrasi teknik pencahayaan proses pengambilan gambar braille seperti terlihat pada Gambar 3.2.

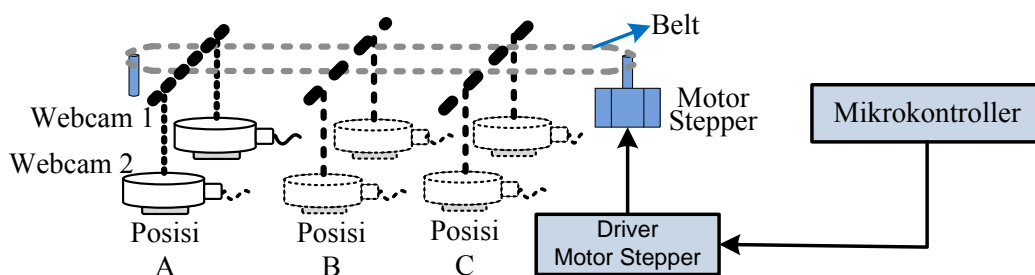


Gambar 3.2 Ilustrasi teknik pencahayaan proses pengambilan gambar braille



Gambar 3.3 Ilustrasi teknik pengambilan gambar braille pada kertas braille dengan 2 webcam dan 3 posisi

Pada Gambar 3.3 dapat diamati ilustrasi teknik pengambilan gambar braille. Dengan ukuran kertas braille lebar 24,8cm dan panjang 30,5cm dan diinginkan semua area kertas braille dapat dicapture oleh webcam. Jika digunakan hanya 1 webcam maka gambar yang dihasilkan kurang baik, cetakan timbul tidak terlihat dengan jelas. Maka digunakan 2 webcam dengan tujuan agar area pengambilan gambar pada kertas braille dapat di bagi menjadi 2 area, dengan demikian maka jarak webcam dengan kertas braille dapat lebih dekat, sehingga gambar hasil capture webcam pada titik-titik timbul pada kertas braille dapat lebih terlihat dengan jelas. Dan area pengambilan gambar pada kertas braille dilakukan dengan 3 posisi yang berbeda, yaitu posisi A, posisi B dan posisi C. Untuk memposisikan webcam kamera pada posisi A, B dan C. Maka ditambahkan rangkaian kontrol posisi dengan digerakkan oleh motor stepper.



Gambar 3.4 Ilustrasi kontrol posisi webcam dengan motor stepper

Pada komputer dibuatkan aplikasi dari Visual Studio 2010 Express, dan menggunakan library dari OpenCV. Adapun perintah capture gambar dengan kamera menggunakan OpenCV adalah sebagai berikut:

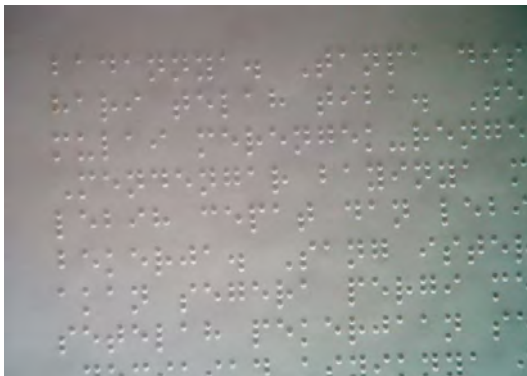
Untuk webcam 1:

```
capture1 = cvCaptureFromCAM(1);  
frame1 = cvQueryFrame( capture1 );
```

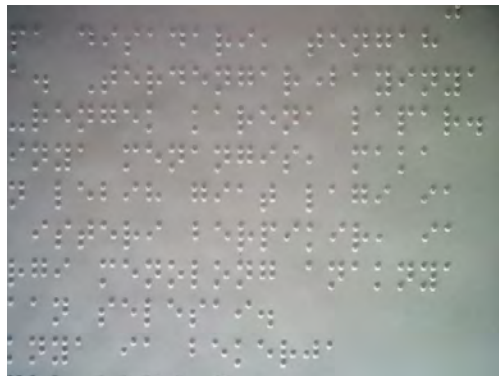
Untuk webcam 2:

```
Capture2 = cvCaptureFromCAM(2);  
Frame2 = cvQueryFrame( capture2 );
```

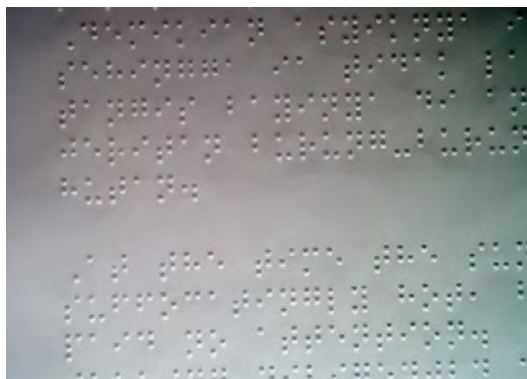
Nilai 1 dan 2 yang ada dalam tanda kurung merupakan nomor id dari camera yang tersedia pada komputer atau laptop yang digunakan. Pada Gambar 3.5 a sampai f adalah contoh hasil pengambilan gambar braille dengan webcam.



(a) Webcam 1 posisi A



(b) Webcam 2 posisi A



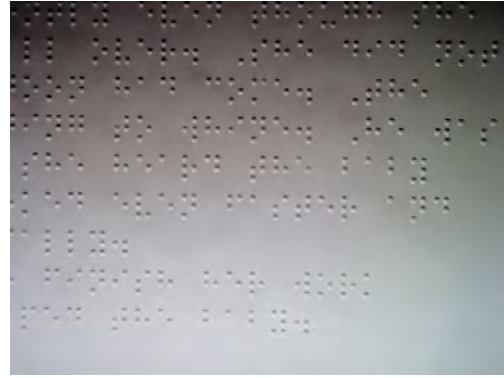
(c) Webcam 1 posisi B



(d) Webcam 2 posisi B



(e) Webcam 1 posisi C



(f) Webcam 2 posisi C

Gambar 3.5 Hasil capture gambar braille dengan webcam

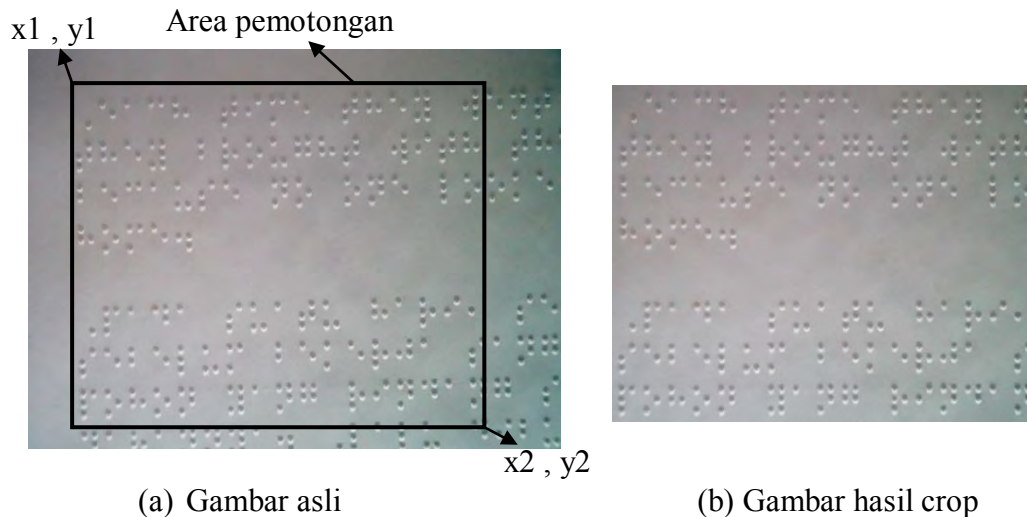
3.2 Pengolahan Gambar Braille

Proses pengolahan gambar braille bertujuan untuk melakukan proses perubahan gambar braille hasil dari capture dari webcam yang masih gambar RGB sampai menjadi gambar biner agar lebih mudah dalam proses pembacaan huruf braille. Dimana proses pengolahan gambar braille ini meliputi:

- Crop Image
- Grayscale
- Thresholding
- Dilatasi
- Erosi

3.2.1 Crop Image

Crop image atau pemotongan gambar dilakukan untuk memotong bagian tepi dari gambar yang tidak diperlukan. Koordinat piksel area pemotongan ditentukan dengan cara manual berdasarkan percobaan berkali-kali sampai didapat nilai koordinat piksel area pemotongan yang sesuai. Karena pengambilan gambar pada satu lembar kertas braille didapat enam gambar, maka penentuan koordinat area pemotongan juga berbeda-beda.



Gambar 3.6 Hasil proses cropping image

Untuk melakukan proses cropping tersebut digunakan fungsi dari OpenCV dengan perintah:

```
cvSetImageROI(image, cvRect(x1, y1, x2, y2))
```

Dimana **image** adalah gambar yang akan diproses, **x1** adalah awal koordinat x, **y1** adalah awal koordinat y, **x2** adalah akhir koordinat x, dan **y2** adalah akhir koordinat y. Pada Gambar 3.6 (a) terlihat pada area diluar garis area pemotongan, yaitu sisi atas dan sisi kiri. Karena tidak diperlukan maka akan dibuang, sedangkan pada sisi kanan dan sisi bawah area yang diluar garis pemotongan adalah area yang masuk pada gambar lain, area tersebut akan diproses tersendiri pada pemrosesan gambar lainnya.

Berdasarkan beberapa kali percobaan dalam menentukan koordinat area cropping agar didapat hasil cropping yang sesuai, maka dapat ditentukan nilai koordinat **x1**, **y1**, **x2** dan **y2** untuk area cropping adalah sebagai berikut:

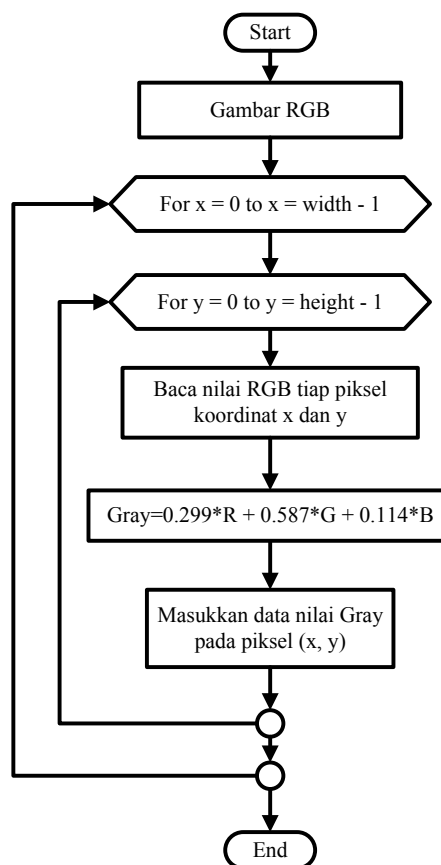
- 1) Gambar webcam1 posisi A = **x1:50 , y1:45 , x2:505 , y2:400**
- 2) Gambar webcam2 posisi A = **x1:80 , y1:15 , x2:550 , y2:455**
- 3) Gambar webcam1 posisi B = **x1:50 , y1:2 , x2:505 , y2:450**
- 4) Gambar webcam2 posisi B = **x1:85 , y1:25 , x2:555 , y2:450**
- 5) Gambar webcam1 posisi C = **x1:50 , y1:15 , x2:510 , y2:445**
- 6) Gambar webcam2 posisi C = **x1:90 , y1:40 , x2:560 , y2:400**

3.2.2 Proses Grayscale

Proses selanjutnya adalah proses grayscale, yaitu merubah gambar dari skala RGB menjadi skala keabuan, sehingga gambar hasil proses grayscale memiliki nilai intensitas warna RGB yang sama. Dalam proses ini gambar hasil proses crop yang akan diproses grayscale. Untuk mendapatkan gambar grayscale bisa digunakan Persamaan 2.1, namun dalam penelitian ini digunakan fungsi yang telah disediakan oleh OpenCV, yaitu dengan perintah:

```
cvCvtColor(src ,bwsrc, CV_RGB2GRAY)
```

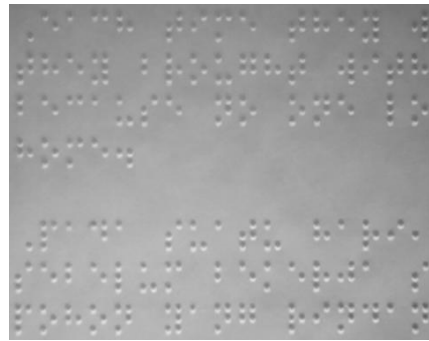
Dimana *src* adalah gambar sumber yang akan diproses, *bwsrc* adalah gambar hasil proses grayscale, *CV_RGB2GRAY* adalah perintah untuk merubah dari RGB ke grayscale. Diagram alir proses grayscale dapat diamati pada Gambar 3.7 dan hasil proses grayscale dapat diamati pada Gambar 3.8.



Gambar 3.7 Diagram alir proses grayscale



(a) Gambar RGB

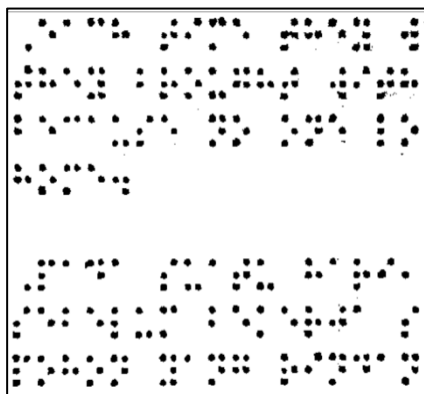


(b) Gambar grayscale

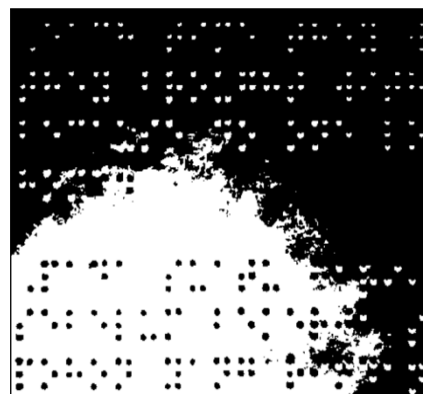
Gambar 3.8 (a) Gambar RGB, (b) Gambar hasil proses grayscale

3.2.3 Proses Thresholding

Dari gambar grayscale selanjutnya dilakukan proses thresholding, proses ini bertujuan untuk memisahkan atau membedakan gambar titik timbul huruf braille dengan gambar area kertas dan akan dihasilkan gambar biner . Gambar biner merupakan gambar yang hanya memiliki nilai skala warna 0 dan 255 atau bisa dikatakan sebagai warna hitam dan putih saja. Pada proses thresholding ini digunakan jenis adaptive threshold, jenis ini berbeda dengan threshold biasa. Yang membedakan adalah dari nilai threshold yang ditetapkan. Kalau threshold biasa menggunakan hanya satu nilai threshold, sedangkan adaptive threshold menggunakan nilai jangkauan minimum dan maksimum.



(a)



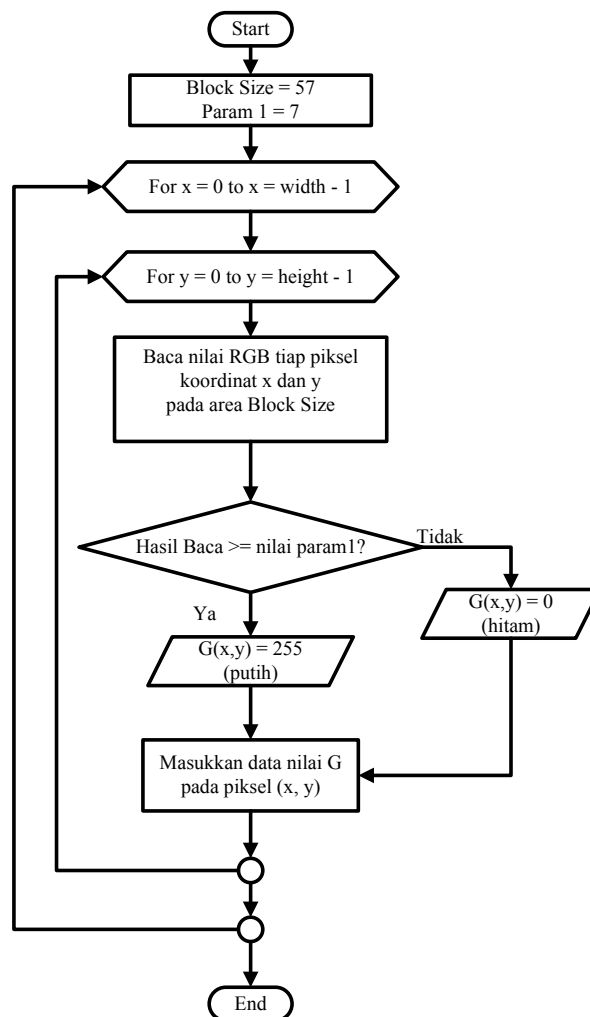
(b)

Gambar 3.9 (a) Hasil adaptive threshold , (b) Hasil general threshold

Untuk melakukan proses thresholding dengan Adaptive Threshold digunakan fungsi yang telah disediakan oleh OpenCV yaitu:

```
cvAdaptiveThreshold(const CvArr* src, CvArr* dst, double
maxValue, int adaptive method=CV_ADAPTIVE_THRESH_GAUSSIAN, int
thresholdType=CV_THRESH_BINARY, int blockSize, double param1 )
```

Pada proses ini nilai `maxValue` diisi 255, `adaptive method` yang digunakan adalah `CV_ADAPTIVE_THRESH_GAUSSIAN`, untuk `thresholdType` dipilih `CV_THRESH_BINARY`, untuk `blockSize` diberikan nilai 57 dan `param1` diberikan nilai 7, namun nilai tersebut bisa dirubah disesuaikan dengan gambar yang akan diproses untuk mendapatkan hasil yang baik.



Gambar 3.10 Diagram alir proses thresholding

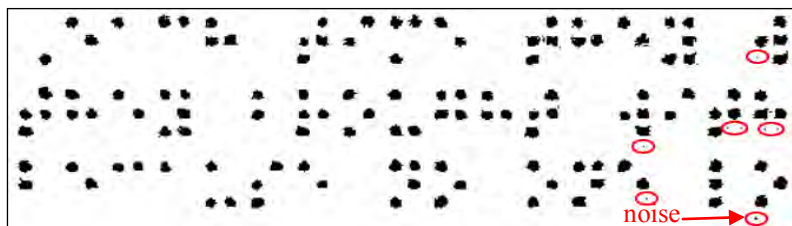
3.2.4 Proses Dilatasi

Sebenarnya proses dilatasi merupakan proses menambahkan atau mempertebal objek pada gambar biner, namun pada perancangan ini proses dilatasi digunakan untuk memperkecil area titik hitam yang merupakan representasi dari titik timbul dari huruf braille. Dilakukan proses ini dengan tujuan untuk menghilangkan noise atau derau yang berupa bintik bintik hitam hasil dari proses thresholding. Dimana bintik-bintik hitam tersebut bukan merupakan titik hitam dari titik timbul dari huruf braille. Dengan dilakukan pengikisan area objek atau memperkecil area titik hitam, maka bintik hitam kecil yang merupakan noise akan hilang dan tersisa bintik hitam besar yang merupakan titik timbul dari huruf braille. Hasil proses dilatasi seperti terlihat pada Gambar 3.11 dan Gambar 3.12.

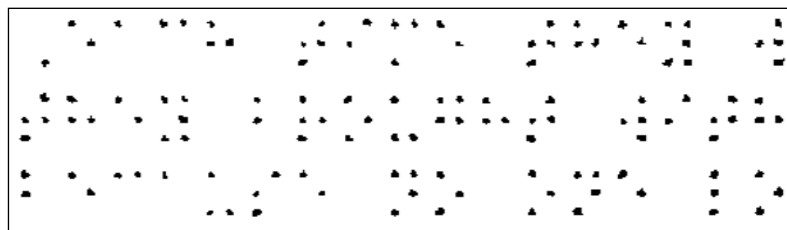
Untuk melakukan proses dilatasi tersebut digunakan fungsi perintah dari OpenCV yaitu `CV_DILATE`. Berikut perintah lengkapnya:

```
cvDilate(src, dst, IplConvKernel* element=NULL, int iterations=1)
```

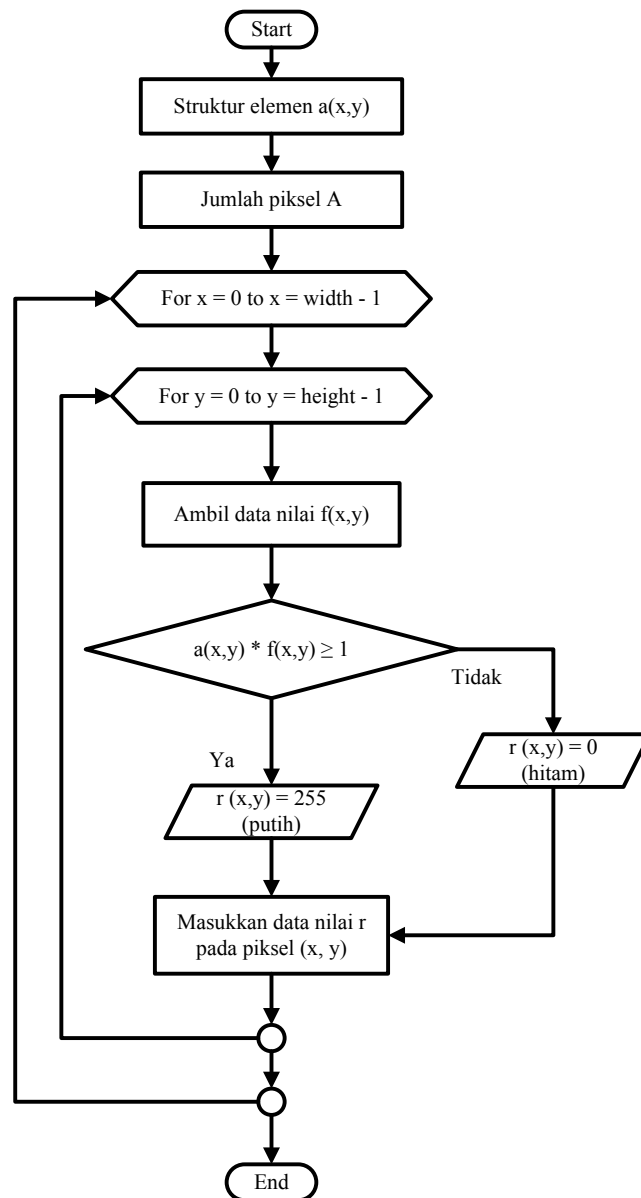
Dimana `src` merupakan gambar yang akan diproses dilatasi, `dst` adalah gambar hasil proses dilatasi, `element` diisi 0 (null) dan `iterations` adalah nilai berapa banyak perulangan proses dilatasi tersebut dilakukan. Jika `iterations` diisi 1, maka proses dilatasi dilakukan 1 kali.



Gambar 3.11 Hasil proses adaptive threshold, masih terdapat noise



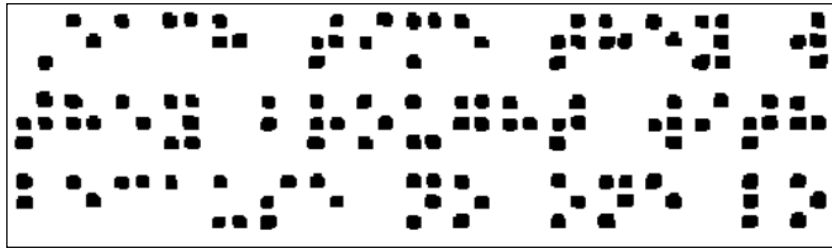
Gambar 3.12 Hasil proses dilatasi (noise hilang)



Gambar 3.13 Diagram alir proses dilatasi

3.2.5 Proses Erosi

Proses erotion sebenarnya merupakan proses untuk mengecilkan ataupun menipiskan objek pada gambar, atau merupakan kebalikan dari proses dilatasi. Namun pada penelitian ini proses erotion digunakan untuk memperbesar kembali titik hitam pada gambar. Seperti terlihat pada Gambar 3.14 dimana hasil erosi titik hitam kembali membesar. Untuk melakukan proses erotion ini digunakan fungsi perintah dari opencv yaitu CV_ERODE.

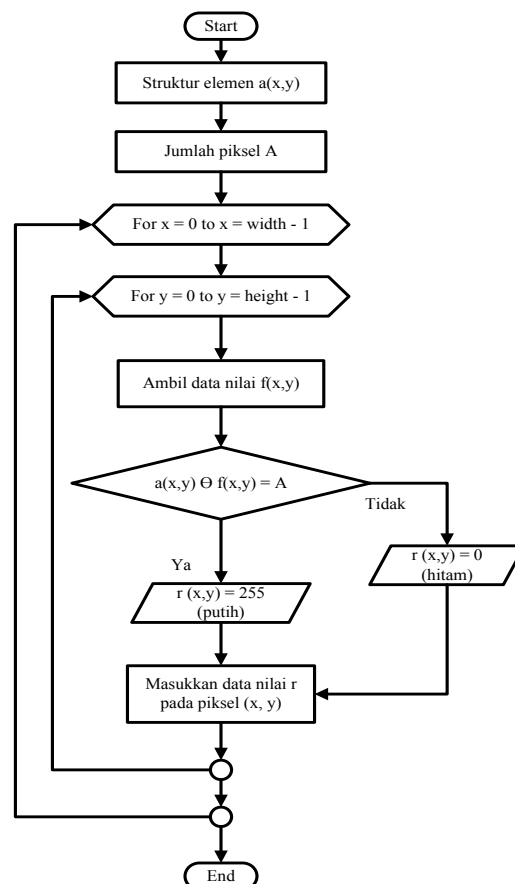


Gambar 3.14 Hasil proses erosi (titik hitam membesar kembali)

Berikut perintah proses erosi menggunakan fungsi dari OpenCV:

`cvErode(src, dst, IplConvKernel* element=NULL, int iterations=1)`

Dimana `src` merupakan gambar yang akan diproses erosi, `dst` adalah gambar hasil proses erosi, `element` diisi 0 (null) dan `iterations` adalah nilai berapa banyak perulangan proses erosi tersebut dilakukan. Jika `iterations` diisi 1, maka proses erosi dilakukan 1 kali.



Gambar 3.15 Diagram alir proses erosi

3.3 Pembacaan Huruf Braille

Untuk dapat melakukan pembacaan huruf braille dengan baik, maka perlu dilakukan proses pembacaan berapa banyak titik hitam yang merupakan representasi dari titik timbul huruf braille yang terdapat dalam gambar braille. Setelah diketahui jumlah titik hitam, kemudian dicari koordinat x dan y tiap titik tersebut, dan ditentukan nilai koordinat rata-rata pada tiap kolom dan baris pada titik-titik hitam yang ada pada gambar. Selanjutnya koordinat tersebut akan dijadikan acuan untuk menentukan area segmentasi dari setiap huruf braille untuk dilakukan pembacaan. Untuk melakukan proses tersebut maka digunakan teknik blob analysis atau deteksi kontur. Kemudian untuk pembacaan huruf braille akan digunakan metode artificial neural network (ANN), data input untuk proses ANN didapat dari pembacaan nilai area segmentasi pada huruf braille. Berikut penjelasan dari proses pembacaan huruf braille, mulai dari proses blob analysis sampai dengan proses ANN. Proses pembacaan dilakukan mulai dari baris yang paling atas menuju ke baris yang paling bawah, dimana dalam tiap baris terdapat sejumlah kolom berdasarkan jumlah titik hitam yang ada pada kolom.

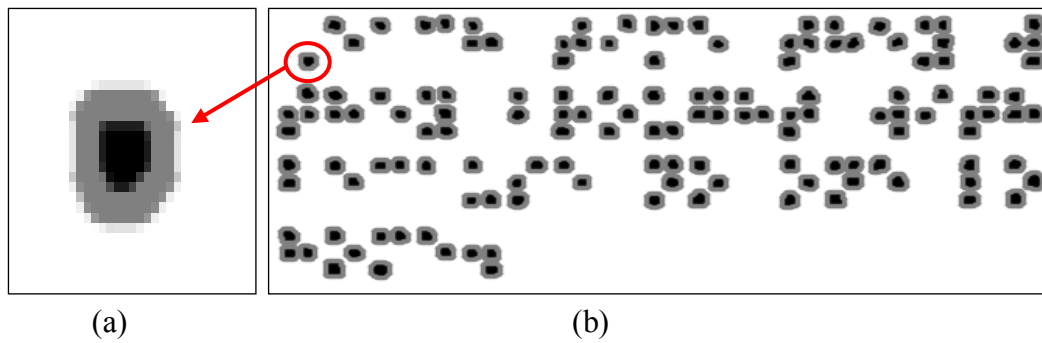
3.3.1 Blob Analysis

Pada proses blob analysis yang bertujuan untuk mendapatkan kontour dari titik-titik dot hitam pada gambar braille, jika kontour dari tiap titik dot hitam tersebut dapat ditemukan maka posisi koordinat tiap titik dot hitam dapat diketahui dan jumlah titik dot hitam pada gambar tersebut dapat dihitung jumlahnya. Untuk melakukan proses ini digunakan fungsi perintah dari OpenCV yaitu `cvFindContour`. Berikut perintah lengkapnya:

```
cvFindContours(src image, dst image, &contours, sizeof  
(CvContour), CV_RETR_LIST)
```

Dimana `src image` adalah gambar sumber yang akan diproses dan `dst image` adalah gambar hasil dari proses blob analysis ini.

Dalam proses blob analysis ini dapat dibaca juga jumlah anggota dari tiap titik dot hitam. Dalam satu titik dot bisa terdapat lebih dari satu anggota. Anggota tersebut merupakan kumpulan piksel yang masuk dalam area titik dot hitam, dan tiap piksel memiliki koordinat x dan y.

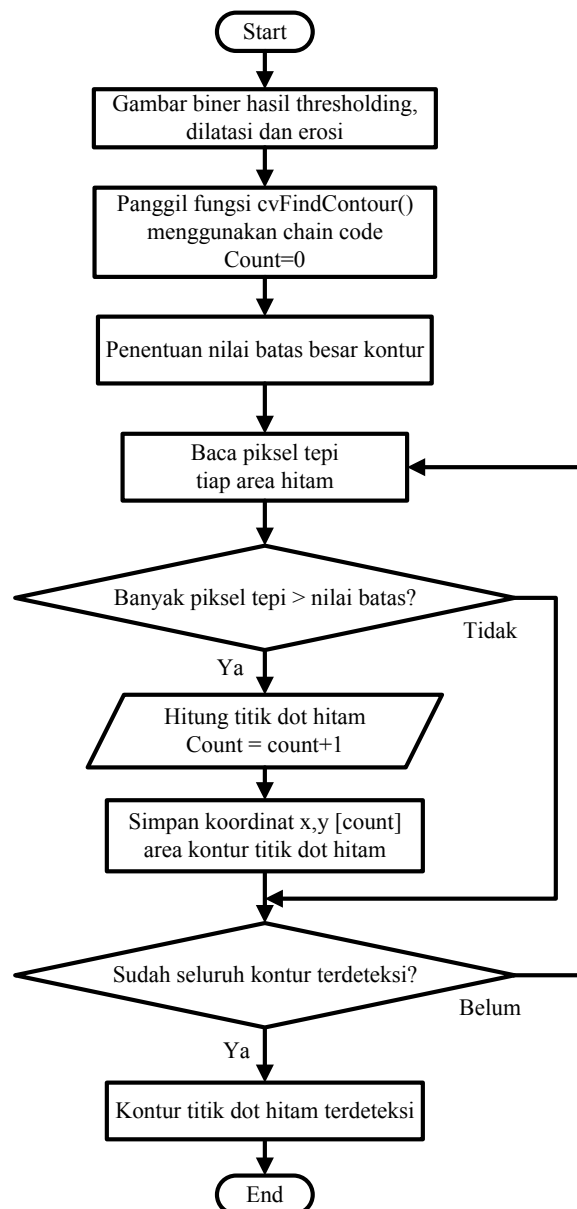


Gambar 3.16 (a) Representasi hasil proses blob analysis pada satu titik dot hitam
(b) Hasil blob analysis keseluruhan titik dot hitam

Pada Gambar 3.16 merupakan gambar hasil proses blob analysis. Pada Gambar 3.16 (a) merupakan contoh dari salah satu titik dot yang terdeteksi, warna hitam pada bagian tengah merupakan titik dot hitam asli, sedangkan pada bagian tepi yang berwarna abu-abu merupakan tanda dari koordinat dari piksel-piksel yang terdeteksi sebagai tepi dari area titik hitam. Dimana piksel-piksel tepi tersebut terdiri atas nilai-nilai koordinat x dan y. Berikut ini adalah contoh data koordinat x,y dari piksel-piksel tepi tersebut pada salah satu titik dot hitam.

```
(1). p->x: 185, p->y: 380   (2). p->x: 189, p->y: 380
(3). p->x: 191, p->y: 382   (4). p->x: 191, p->y: 385
(5). p->x: 189, p->y: 387   (6). p->x: 186, p->y: 387
(7). p->x: 185, p->y: 386   (8). p->x: 184, p->y: 386
(9). p->x: 183, p->y: 385   (10). p->x: 183, p->y: 382
Banyak piksel tepi: 10
```

Dari data diatas dapat diketahui bahwa pada contoh satu titik dot hitam tersebut terdapat 10 koordinat piksel tepi. Jumlah banyaknya piksel pada setiap titik area dot hitam dapat bervariasi tergantung luas dari area titik dot hitam tersebut. Semakin luas atau semakin besar titik dot hitam, maka semakin banyak koordinat piksel tepi dari area tersebut. Koordinat piksel tepi setiap titik area disimpan dalam suatu variabel, yang nantinya digunakan kembali untuk penentuan titik koordinat tengah dari masing-masing area titik dot hitam tersebut. Untuk lebih jelasnya proses blob analysis ini dapat diamati pada Gambar 3.17 yang merupakan diagram alir dari proses blob analysis ini.



Gambar 3.17 Diagram alir proses blob analysis

3.3.1.1 Membaca Banyak Titik Dot Hitam

Proses membaca banyak titik dot hitam pada area gambar dilakukan untuk mengetahui berapa banyak titik dot hitam yang ada. Jika terjadi kesalahan dalam menghitung banyak titik dot hitam, maka akan mengakibatkan kesalahan pembacaan data koordinat sehingga proses pengenalan huruf braille akan mengalami kesalahan. Cara menghitung banyak titik dot hitam yaitu dengan

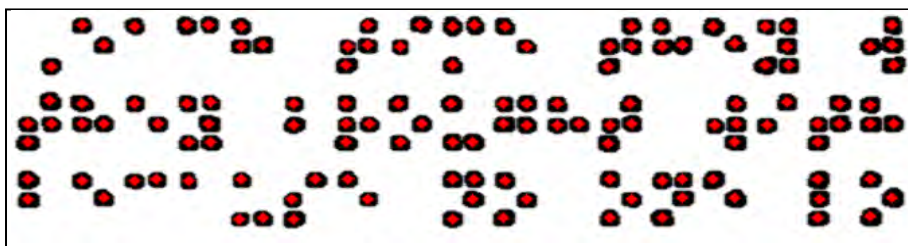
menentukan nilai batas kumpulan banyak koordinat piksel tepi pada area titik dot hitam tersebut. Semisal, jika ditentukan nilai batas adalah 5, maka area titik dot hitam yang memiliki lebih dari 5 koordinat piksel tepi maka area titik dot tersebut dihitung sebagai satu titik dot. Sedangkan area titik hitam yang koordinat piksel tepinya kurang dari 5, maka area tersebut tidak dianggap sebagai titik dot hitam, melainkan dianggap sebagai bintang hitam yang bisa diabaikan. Berikut perintah untuk menghitung banyak titik dot hitam:

```
count=0;
for( CvSeq* c=contours; c!=NULL; c=c->h_next )
{
    tot=c->total;
    if(tot>5){count++;};
}
```

Penjelasan dari perintah diatas adalah: program melakukan proses berulang-ulang untuk mengecek apakah nilai banyak anggota dari satu titik hitam melebihi dari nilai batas yang ditentukan. Jika nilai `c->total` lebih besar dari nilai batas yang ditentukan maka, hitung banyak titik dot ditambah 1 nilainya, dalam program tersebut dinamakan count.

3.3.1.2 Mencari Koordinat Setiap Titik Dot Hitam

Dari contoh data kumpulan titik koordinat piksel tepi yang terdiri atas 10 data tersebut, selanjutnya dicari satu koordinat titik tengah. Cara menghitungnya yaitu dicari nilai minimum dan maksimum dari koordinat x dan koordinat y dari data koordinat 10 piksel tepi, setelah diketahui nilai minimum dan maksimum kemudian nilai tersebut dijumlahkan dan kemudian dibagi 2. Sehingga akan dapat dihitung nilai tengah antar nilai minimum dengan nilai maksimum.



Gambar 3.18 Hasil pembacaan titik tengah tiap dot

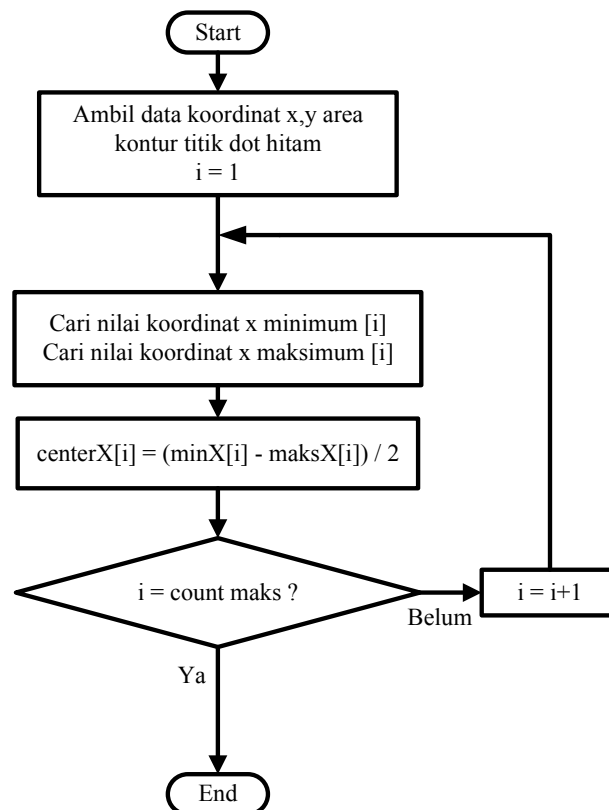
Pada Gambar 3.18 dapat diamati hasil dari pembacaan koordinat tengah dari tiap titik dot hitam, dengan ditandai titik warna merah. Cara mencari koordinat titik tengah koordinat x digunakan Persamaan (3.1) berikut ini:

$$CenterX = \frac{minX + maxX}{2} \quad (3.1)$$

Dimana:

centerX = hasil pembacaan titik tengah koordinat x
minX = nilai minimum koordinat x
maxX = nilai maksimum koordinat x

Untuk lebih jelas pada proses mencari titik tengah koordinat x setiap titik dot hitam dapat diamati pada Gambar 3.19 berikut ini.



Gambar 3.19 Diagram alir proses mencari titik tengah koordinat x

Sedangkan untuk mencari koordinat titik tengah koordinat y digunakan Persamaan (3.2) berikut ini:

$$CenterY = \frac{minY + maxY}{2} \quad (3.2)$$

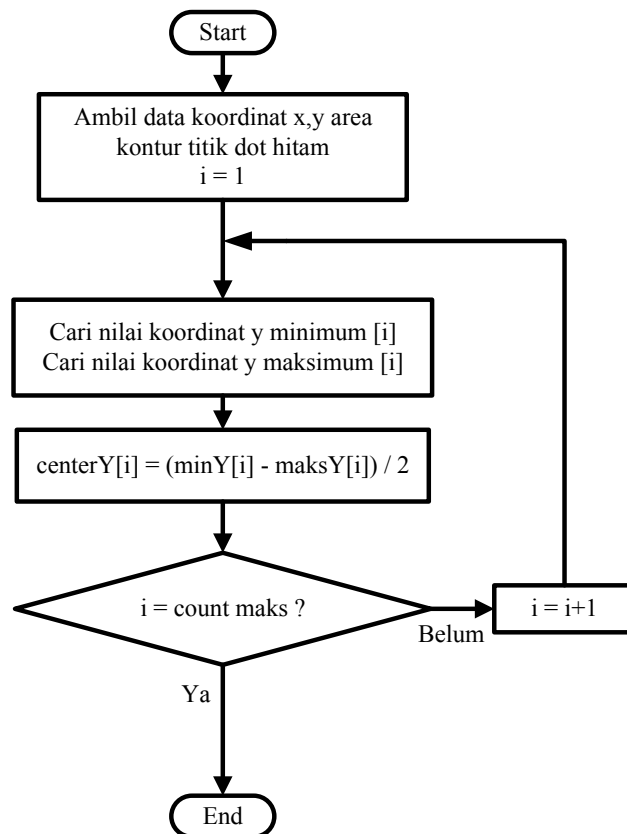
Dimana:

centerY = hasil pembacaan titik tengah koordinat y

minY = nilai minimum koordinat y

maxY = nilai maksimum koordinat y

Untuk lebih jelas pada proses mencari titik tengah koordinat y setiap titik dot hitam dapat diamati pada Gambar 3.20 berikut ini.



Gambar 3.20 Diagram alir proses mencari titik tengah koordinat y

3.3.1.3 Mencari Koordinat Rata-rata Setiap Titik Dot pada Baris dan Kolom

Setelah nilai koordinat tengah tiap titik dot dapat diketahui, selanjutnya dicari nilai rata-rata tengah pada tiap baris dan kolom. Mengapa demikian, karena dalam satu baris maupun dalam satu kolom belum tentu memiliki koordinat x dan y yang sama. Sehingga perlu dicari nilai rata-rata dari nilai koordinat tersebut. Pada Gambar 3.21 dapat diamati hasil dari proses mencari koordinat rata-rata pada baris dan kolom, dengan ditandai garis merah pada baris dan kolom. Kelihatan semua titik dot hitam mengenai garis tanda merah tersebut, sehingga tidak ada titik dot hitam yang terlewatkan.

Untuk mencari nilai rata-rata koordinat x digunakan Persamaan (3.3) dan untuk mencari rata-rata koordinat y tiap baris dan kolom digunakan Persamaan (3.4). Berikut adalah kedua persamaan tersebut:

$$rataX_k = \sum_{k=1}^k \sum_{n=1}^n (rataX_k + CenterX_n)/n \quad (3.3)$$

Dimana:

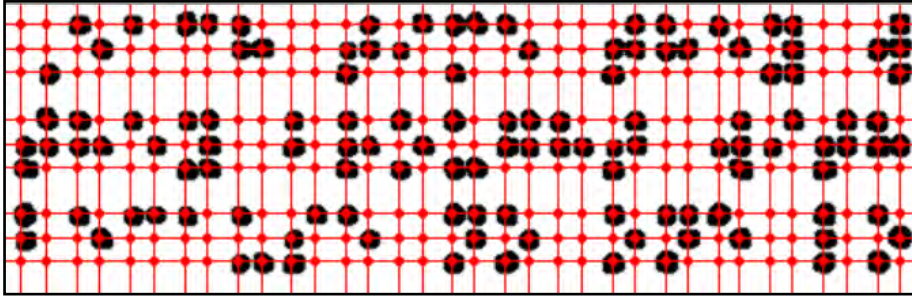
- $rataX_k$ = Nilai rata-rata koordinatX tiap kolom
- $CenterX_n$ = Nilai koordinat centerX tiap kolom
- k = Banyak kolom
- n = Banyak titik dot dalam tiap kolom

$$rataY_b = \sum_{b=1}^b \sum_{n=1}^n (rataY_b + CenterY_n)/n \quad (3.4)$$

Dimana:

- $rataY_b$ = Nilai rata-rata koordinatY tiap baris
- $CenterY_n$ = Nilai koordinat centerY tiap baris
- b = Banyak baris
- n = Banyak titik dot dalam tiap baris

Hasil mencari koordinat rata-rata tiap titik dot pada baris dan kolom dapat diamati pada Gambar 3.21.



Gambar 3.21 Hasil mencari koordinat rata-rata titik dot pada baris dan kolom

Dengan sudah diketahui koordinat rata-rata titik dot hitam tiap baris dan kolom tersebut maka akan memudahkan dalam proses segmentasi dan pengenalan huruf braille selanjutnya.

3.3.1.4 Menentukan Segmentasi Area

Dari data koordinat rata-rata yang sudah didapat pada tiap baris dan kolom, selanjutnya dijadikan sebagai acuan proses segmentasi. Proses segmentasi bertujuan menentukan area untuk membaca data tiap satu huruf braille saja. Karena pada huruf braille terdiri atas 2 kolom dan 3 baris, maka penentuan area segmentasi adalah berdasarkan nilai rata-rata koordinat x dan y tiap-tiap 6 titik koordinat x,y. Perhitungan untuk menentukan area segmentasi digunakan persamaan (3.5) berikut:

$$ASeg_i = \sum_{b=1}^b \sum_{k=1}^k [i = i + 1] \left(\sum_{m=b}^{b+3} \sum_{n=k}^{k+2} (rataX_n, rataY_m) \right) \quad (3.5)$$

Dimana:

$ASeg_i$ = Area Segmentasi tiap huruf braille

b = Banyak baris

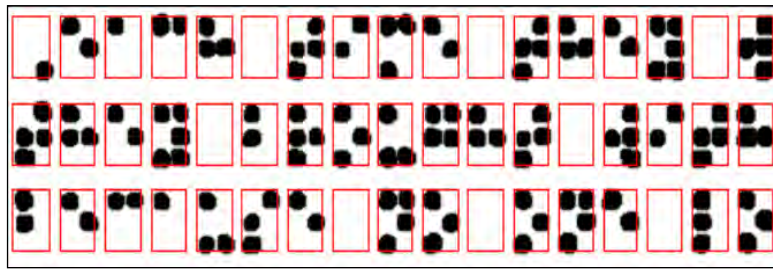
k = Banyak kolom

m = Hitung 3 baris untuk satu huruf braille

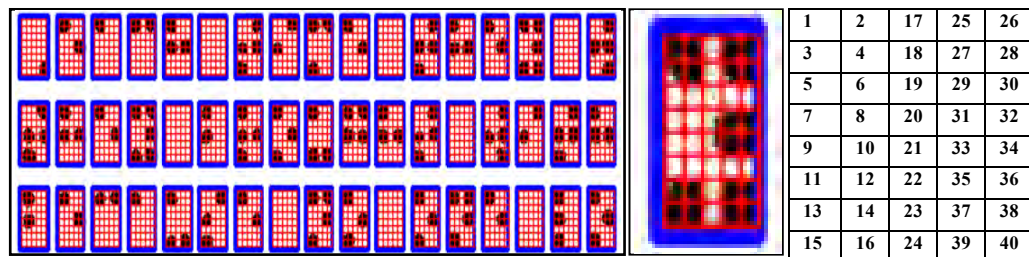
n = Hitung 2 kolom untuk satu huruf braille

$rataX_n$ = nilai rata-rata koordinat X pada kolom ke n

$rataY_m$ = nilai rata-rata koordinat Y pada baris ke m



Gambar 3.22 Hasil proses segmentasi area tiap satu karakter huruf braille



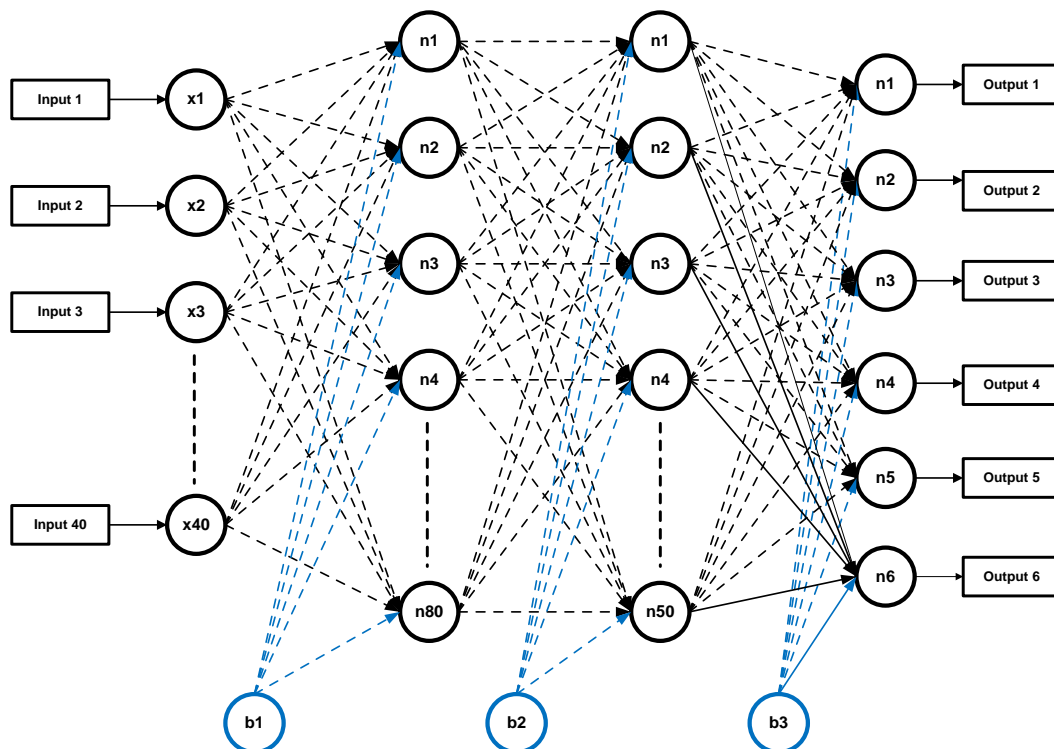
Gambar 3.23 Segmentasi kecil sebanyak 40 segmentasi pada area satu huruf braille

Hasil proses segmentasi dapat diamati pada Gambar 3.22. Area segmentasi per satu huruf braille sudah bisa ditandai dengan garis merah. Setelah ditentukan area segmentasi tiap huruf, dilakukan segmentasi yang lebih kecil lagi pada area tiap huruf sebanyak 40 segmentasi kecil.

Pada Gambar 3.23 terlihat bahwa pada area satu huruf braille disegmentasi lagi menjadi area yang kecil-kecil lagi, yaitu terdapat 40 area segmentasi kecil pada area satu huruf. Segmentasi tersebut terdiri atas 8 area segmentasi pada baris dan 5 area segmentasi pada kolom, sehingga total segmentasi kecil tersebut adalah 40 segmentasi area kecil. Dimana 40 area segmentasi kecil tersebut selanjutnya akan dibaca nilai skala dari pikselnya, bernilai 0 atau 255 (hitam atau putih). Sehingga dari hasil pembacaan tersesbut akan didapat 40 nilai data, dan data inilah yang akan dijadikan nilai data input proses pengenalan huruf braille dengan artificial neural network.

3.3.2 Artificial Neural Network

Selanjutnya dilakukan pengenalan huruf braille untuk diterjemahkan ke teks dengan menggunakan metode Artificial Neural Network (ANN). Rancangan sistem ANN pada penelitian ini didesain atas 1 input layer, 2 hidden layer dan 1 output layer. Pada input layer sebagai masukan 40 data input yang telah didapat dari hasil pembacaan nilai skala piksel pada segmentasi area kecil. Pada hidden layer 1 didesain dengan neuron sebanyak 80 node, pada hidden layer 2 didesain dengan neuron sebanyak 50 node, dan pada output layer didesain dengan 6 neuron. Pada layer output dipilih dengan 6 neuron dengan maksud agar pada proses ANN ini akan menghasilkan data keluaran dengan kapasitas 6 bit. Dimana kapasitas 6 bit tersebut akan dapat menampung nilai kombinasi dari 0 sampai 33, karena jumlah data huruf alphabet adalah 26 (a-z), ditambah tanda spasi, koma “,” , titik “.” , tanda tanya “?” , tanda angka, tanda kapital, strip tengah “-” dan tanda seru “!” . Jadi total keseluruhan adalah 34 data huruf. Rancangan topologi dari ANN terlihat pada Gambar 3.24 berikut.



Gambar 3.24 Topologi jaringan ANN pada sistem pengenalan huruf braille

3.3.2.1 Input Artificial Neural Network

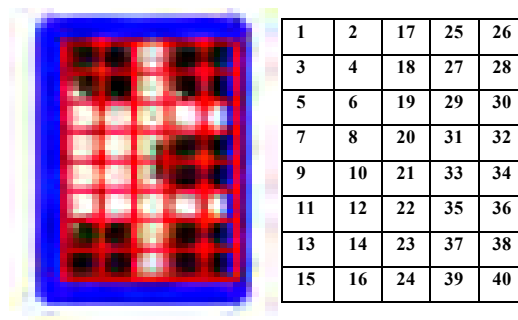
Data input pada ANN terdiri dari 40 data input. Data input tersebut didapatkan dari hasil pembacaan nilai setiap area segmentasi kecil pada setiap area segmentasi satu huruf braille, dimana pada setiap satu area segmentasi huruf braille terdiri atas 40 segmentasi kecil. Contoh area segmentasi dapat diamati pada Gambar 3.25. Nilai data input berupa data biner yang hanya bernilai “0” atau “1”, penentuan nilai data input bernilai 0 atau 1 diperoleh dari pembacaan nilai intensitas piksel dari area tersebut. Jika nilai intensitas piksel bernilai 255 atau berwarna putih maka diberi nilai 0, sedangkan jika nilai intensitas piksel bernilai 0 atau berwarna hitam maka diberi nilai 1.

Pada satu area segmentasi kecil terdapat lebih dari satu piksel, maka penentuan nilai input pada satu segmentasi kecil tersebut berdasarkan jumlah yang terbanyak antara piksel berwarna hitam dan piksel berwarna putih. Jika piksel berwarna hitam lebih banyak dari piksel yang berwarna putih maka pada area segmentasi kecil tersebut bernilai 1, sedangkan jika piksel berwarna putih lebih banyak daripada piksel berwarna hitam maka pada area segmentasi kecil tersebut bernilai 0. Penentuan nilai data input digunakan Persamaan (3.6) dan Persamaan (3.7) berikut ini:

$$\begin{aligned} & \text{hitam} = 0; \text{putih} = 0; \\ & \text{baca} = \begin{cases} \text{hitam} = \text{hitam} + 1, & \text{jika } f(x, y) = 0 \\ \text{putih} = \text{putih} + 1, & \text{jika } f(x, y) = 255 \end{cases} \end{aligned} \quad (3.6)$$

$$\text{nilai input} = \begin{cases} 1, & \text{jika } \text{hitam} > \text{putih} \\ 0, & \text{jika } \text{hitam} < \text{putih} \end{cases} \quad (3.7)$$

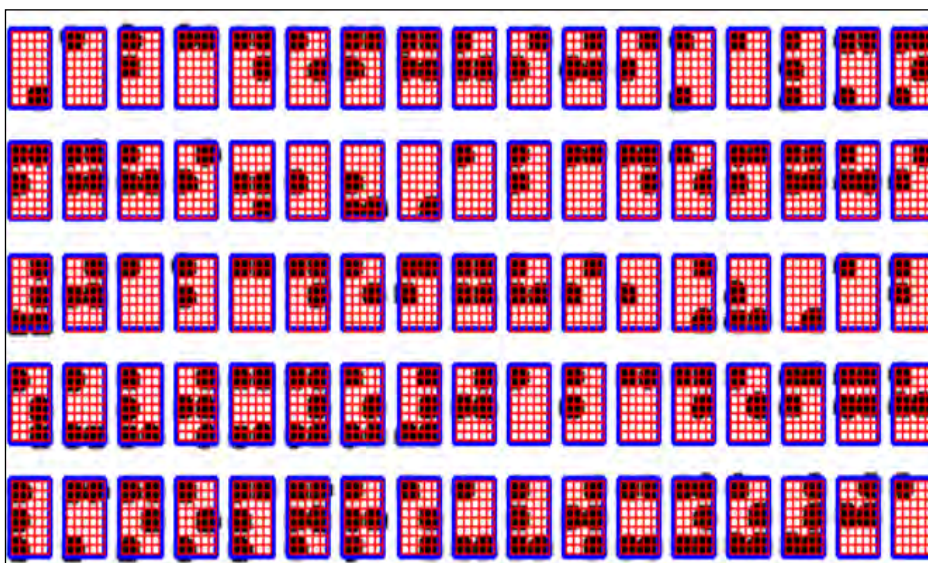
Pembacaan nilai input tersebut dilakukan dari area segmentasi kecil nomor 1 sampai dengan nomor 40, sehingga akan didapatkan suatu nilai kondisi dari satu area segmentasi satu huruf braille yang diwakili oleh 40 data input hasil pembacaan. Proses pembacaan data input tersebut diulang-ulang untuk setiap area segmentasi huruf braille lainnya, termasuk juga untuk area segmentasi huruf “spasi”.



Gambar 3.25 Area segmentasi kecil pada satu huruf braille

Pengambilan data input untuk proses pembelajaran dilakukan langsung pada gambar huruf braille pada setiap segmentasi satu huruf. Gambar yang digunakan untuk mengambil data input untuk proses pembelajaran pada ANN merupakan cetakan dari karakter huruf A sampai Z, tanda spasi, koma, titik, tanda tanya, tanda angka serta tanda kapital. Pengambilan data dilakukan dengan lima macam variasi kemiringan dari gambar huruf braille. Lima macam variasi kemiringan tersebut adalah dengan tingkat kemiringan sebesar 0° , 0.5° , 1° , -0.5° dan -1° .

Pada Gambar 3.26 menunjukkan area segmentasi pengambilan data input, pada gambar tersebut memiliki derajat kemiringan 0 derajat atau kondisi tegak lurus, Sehingga terlihat area segmentasi pada huruf braille tepat semua.



Gambar 3.26 Pengambilan data input pada kemiringan 0 derajat

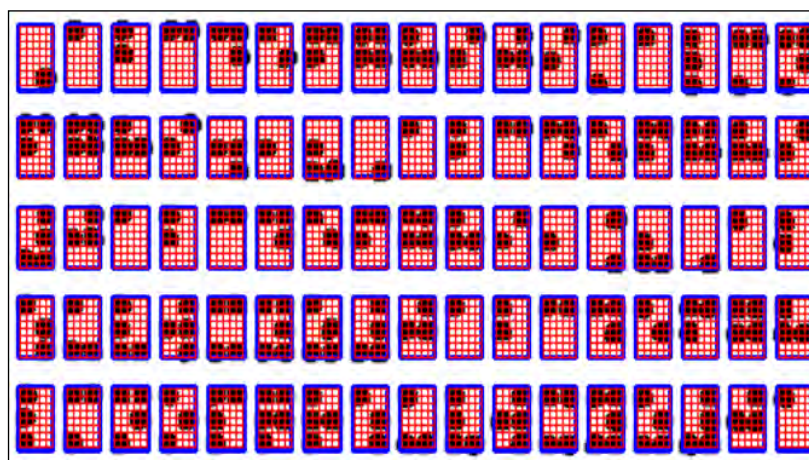
Pada Tabel 3.1 merupakan hasil data input yang diperoleh dari proses pembacaan area segmentasi pada setiap huruf braille dengan derajat kemiringan gambar sebesar 0 derajat (tegak lurus).

TABEL 3.1 Data hasil pengambilan data input dari gambar braille untuk proses pembelajaran pada ANN dengan kemiringan 0 derajat.

<p align="center">Data input ANN pada proses pembelajaran.</p> <p align="center">Gambar huruf braille dengan kemiringan 0° (Nol derajat)</p>		
No.	Data Input (D1 – D40)	Karakter
1.	000000000000000000000000000000000000	SPASI
2.	111111000000000011000000000000000000	A
3.	111111111111000011011000000000000000	B
4.	11110000000000001100000011110000000000	C
5.	111111000000000011011000111111111110000	D
6.	111111000000000001011000000001111110000	E
7.	1111111111111000011011000111111000000000	F
8.	11111111111110000110110001111101111110000	G
9.	111100111111000000011100000011111110000	H
10.	000011111100000011011000111100000000000	I
11.	0000001111100000110110001111001111100000	J
12.	111100000000111100000000000000000000000	K
13.	111100111110111100000000000000000000000	L
14.	111111000000111111000001111111000000000	M
15.	1111110000001111110111001111111111110000	N
16.	11111100000011111111110110000001111110000	O
17.	111111111111111111111111111000000000000	P
18.	11111111111111111111011101111111111110000	Q
19.	11111111111111111111111110000001111010000	R
20.	00001111111111111111011011111110000000000	S
21.	00001111111111111111011011111111111000000	T
22.	1111010000001111110000110000000000001111	U

Data input ANN pada proses pembelajaran. Gambar huruf braille dengan kemiringan 0° (Nol derajat)		
No.	Data Input (D1 – D40)	Karakter
23.	1111111111111111101111100000000000001111	V
24.	00001111111110000110111011111111111111	W
25.	1111110000001111110000111111010000001111	X
26.	11111100000011111100101111111111111111	Y
27.	111101000000111111011011000000111111111	Z
28.	0000000000000000000000110000000000001111	KAPITAL
29.	0000000000001111000000111111011111011111	ANGKA
30.	000000111111000000011001000000111111111	TITIK “. “
31.	000000111111111100001011000000000011111	TANYA “ ? “
32.	000000111111000000001000000000000000000	KOMA “ , “
33.	0000111111111111001110110000111111110000	SERU “ ! “
34.	000011111111111100011011000000000011111	STRIP “ – “

Selanjutnya dilakukan juga pengambilan data input pada area segmentasi setiap huruf braille dengan kemiringan gambar sebesar 0,5° (setengah derajat). Seperti terlihat pada Gambar 3.27 berikut:.



Gambar 3.27 Pengambilan data input pada kemiringan 0,5 derajat

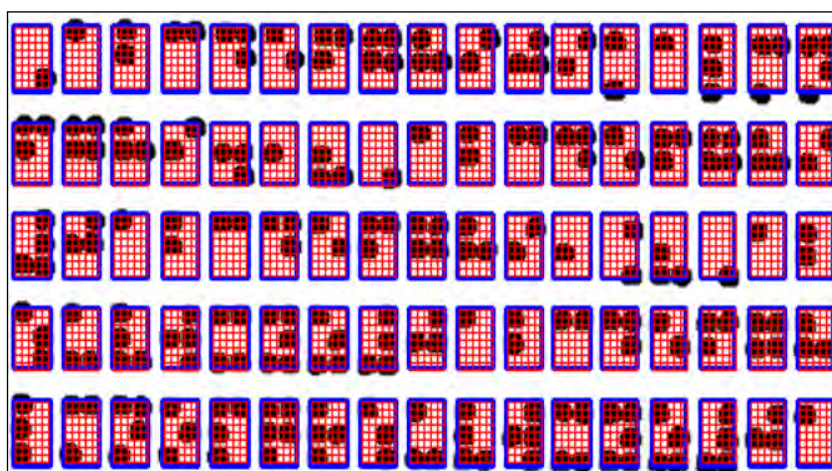
Pada Tabel 3.2 merupakan hasil data input yang diperoleh dari proses pembacaan area segmentasi pada setiap huruf braille dengan derajat kemiringan gambar sebesar 0,5 derajat.

TABEL 3.2 Data hasil pengambilan data input dari gambar braille untuk proses pembelajaran pada ANN dengan kemiringan setengah derajat.

Data input ANN pada proses pembelajaran.		
Gambar huruf braille dengan kemiringan 0,5° (Setengah derajat)		
No.	Data Input (D1 – D40)	Karakter
1.	000000000000000000000000000000000000	SPASI
2.	111100000000000011000000000000000000	A
3.	111111111100000011111000000000000000	B
4.	11110000000000001100000011110000000000	C
5.	11110000000000001100000011111111100000	D
6.	11110000000000001100000000001111110000	E
7.	11111111111000011011000111100000000000	F
8.	11111111111000011011000111111111110000	G
9.	111100111111000011011000000011111110000	H
10.	00001111110000001101100011110000000000	I
11.	000000111110000011011000111100111110000	J
12.	11111000000011111100000100000000000000	K
13.	11111011111111110000100100000000000000	L
14.	11111100000011111110000111111100000000	M
15.	11111100000011111101110111111111110000	N
16.	11110000001111111011011000011111110000	O
17.	11111111110111111111111111100000000000	P
18.	11110011110111111101111111100111110000	Q
19.	1111011111011111111111111000001111100000	R
20.	00001111111111110011101111110000000000	S
21.	00000011111111110001101111111111110000	T
22.	1111000000001111110000110000000000001111	U

Data input ANN pada proses pembelajaran.		
Gambar huruf braille dengan kemiringan 0,5° (Setengah derajat)		
No.	Data Input (D1 – D40)	Karakter
23.	11110111111111111111111110000000000001111	V
24.	0000011111110000001111001111011111111111	W
25.	1111110000001111111000111111010000001111	X
26.	1111110000001111111000111111011111111111	Y
27.	1111010000001111111000110000001111111111	Z
28.	0000000000000000000000000000000000111111	KAPITAL
29.	0000000000001111000000111111011111011111	ANGKA
30.	0000111111000000000110000000111111111111	TITIK “ . “
31.	0000111111111111000110110000000000111111	TANYA “ ? “
32.	0000111111000000000110000000000000000000	KOMA “ , “
33.	0000111111111111001110100000111111000000	SERU “ ! “
34.	00000011111111111000000110000000000001111	STRIP “ – “

Selanjutnya dilakukan juga pengambilan data input pada area segmentasi setiap huruf braille dengan kemiringan gambar sebesar 1° (satu derajat). Seperti terlihat pada Gambar 3.28 berikut:.



Gambar 3.28 Pengambilan data input pada kemiringan 1 derajat

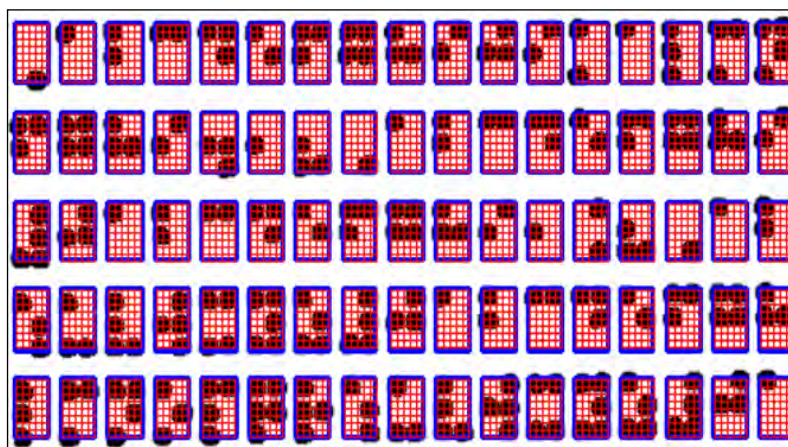
Pada Tabel 3.3 merupakan hasil data input yang diperoleh dari proses pembacaan area segmentasi pada setiap huruf braille dengan derajat kemiringan gambar sebesar 1 derajat.

TABEL 3.3 Data hasil pengambilan data input dari gambar braille untuk proses pembelajaran pada ANN dengan kemiringan 1 derajat.

[illegible]

Data input ANN pada proses pembelajaran. Gambar huruf braille dengan kemiringan 1° (Satu derajat)		
No.	Data Input (D1 – D40)	Karakter
23.	1111011111111111111111111010001010100111	V
24.	00000011111100000000111000111011111110111	W
25.	010101000000001111110000111111110000001111	X
26.	0111110000000011111100001111111110111110011	Y
27.	01010100000000111111000011010100111110011	Z
28.	000000000000000000000000000000000000111111	KAPITAL
29.	00000000000000110000000010101010001010011	ANGKA
30.	000000111100000000001100000000001111111111	TITIK “. “
31.	000000111111111100001111100000000000001111	TANYA “ ? “
32.	000000111111000000001100000000000000000000	KOMA “ , “
33.	0000111111111111001110100000111111000000	SERU “ ! “
34.	000000101111111110000001100000000000001011	STRIP “ – “

Selanjutnya dilakukan juga pengambilan data input pada area segmentasi setiap huruf braille dengan kemiringan gambar sebesar $-0,5^\circ$ (negatif setengah derajat). Seperti terlihat pada Gambar 3.29 berikut:.



Gambar 3.29 Pengambilan data input pada kemiringan $-0,5$ derajat

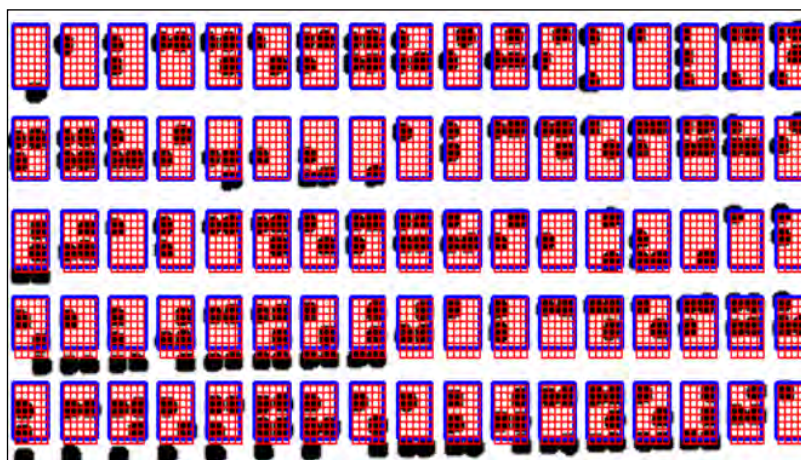
Pada Tabel 3.4 merupakan hasil data input yang diperoleh dari proses pembacaan area segmentasi pada setiap huruf braille dengan derajat kemiringan gambar sebesar -0,5 derajat.

TABEL 3.4 Data hasil pengambilan data input dari gambar braille untuk proses pembelajaran pada ANN dengan kemiringan -0,5 derajat.

Data input ANN pada proses pembelajaran.		
Gambar huruf braille dengan kemiringan -0,5° (Negatif Setengah derajat)		
No.	Data Input (D1 – D40)	Karakter
1.	000000000000000000000000000000000000	SPASI
2.	111111000000000000000000000000000000	A
3.	111111111111000000000000000000000000	B
4.	1111110000000000111000001111110000000000	C
5.	1111110000000000111111001111111111110000	D
6.	11111100000000000000111000000001111110000	E
7.	1111101111110000111000001111100000000000	F
8.	1111111111110000111111001111101111110000	G
9.	1111101111110000000111000000001111110000	H
10.	0000101111100000110000001111000000000000	I
11.	0000001111100000110111001111001111100000	J
12.	1110000000101111000000000000000000000000	K
13.	1111000000000000000000000000000000000000	L
14.	1110101111001111000000000000000000000000	M
15.	1111000000101111110000001111000000000000	N
16.	1111111100000011011011100000000011111100	O
17.	1111111111111111011011001111111100000000	P
18.	111111111111111101101101111110011111100	Q
19.	1111110111111111011011010000001111110100	R
20.	0000001111111111111100000111111000000000	S
21.	00001111111111111111110001111111111000000	T
22.	1111110000000011000000010000000000001011	U

Data input ANN pada proses pembelajaran.		
Gambar huruf braille dengan kemiringan $-0,5^\circ$ (Negatif Setengah derajat)		
No.	Data Input (D1 – D40)	Karakter
23.	111111111111001100000000100000000000001111	V
24.	0000001111110000111111011111111111111111	W
25.	111111000000111111110001111111110000001111	X
26.	1111110000001111111111111111111111111111	Y
27.	1111110000001111000111110000001111111111	Z
28.	000000000000000000000000110000000000001111	KAPITAL
29.	0000000000001111110110111111111111111111	ANGKA
30.	0000001111110000111010001111110000000000	TITIK “. “
31.	0000001111110000000010000000000000000000	TANYA “ ? “
32.	0000001111110000000111110000001111111111	KOMA “ , “
33.	0000111111111111000110110000101111110000	SERU “ ! “
34.	0000111111111111001111110000000000111111	STRIP “ – “

Yang terakhir dilakukan juga pengambilan data input pada area segmentasi setiap huruf braille dengan kemiringan gambar sebesar -1° (negatif satu derajat). Seperti terlihat pada Gambar 3.30 berikut:.



Gambar 3.30 Pengambilan data input pada kemiringan -1 derajat

Pada Tabel 3.5 merupakan hasil data input yang diperoleh dari proses pembacaan area segmentasi pada setiap huruf braille dengan derajat kemiringan gambar sebesar -1 derajat.

TABEL 3.5 Data hasil pengambilan data input dari gambar braille untuk proses pembelajaran pada ANN dengan kemiringan -1 derajat.

Data input ANN pada proses pembelajaran.		
Gambar huruf braille dengan kemiringan -1° (Negatif Satu derajat)		
No.	Data Input (D1 – D40)	Karakter
1.	000000000000000000000000000000000000	SPASI
2.	001111110000000000000000000000000000	A
3.	101111111111110000000000000000000000	B
4.	101111100000000001110000101111100000000	C
5.	0011111000010000011111100011111011111100	D
6.	0011111100000000000011100000000011111100	E
7.	1011110011111000111000001011110000000000	F
8.	1011111111111000111111101011101011111000	G
9.	1011101111110000000111000000001111110000	H
10.	0101001011100000111000001011100000000000	I
11.	0101001111110000111111001010101011100000	J
12.	10100000000010100000000000000000000000	K
13.	10101010100010100000000000000000000000	L
14.	1111000000001111110000001010000000000000	M
15.	1111010101101111111110001010101111000000	N
16.	0010111111000000000001110000000000111111	O
17.	0010111111111111101110000001011111100000	P
18.	0010111110111111011101110010111111111111	Q
19.	00111111111111111000011100000000011111100	R
20.	0101011111111011111000001111110000000000	S
21.	0000111111111111111110001111111111000000	T
22.	0011111100000010000000010000000000000011	U

Data input ANN pada proses pembelajaran. Gambar huruf braille dengan kemiringan -1° (Negatif Satu derajat)		
No.	Data Input (D1 – D40)	Karakter
23.	001111111111111100000000100000000000000011	V
24.	0000000011111100111111111011111111111111	W
25.	1011111000000011111100001111111000000011	X
26.	1011111000000011111011011111111111111111	Y
27.	1111110001010011000111010000001111111111	Z
28.	0000000000000000000000001000000000000010	KAPITAL
29.	0001000000001111111111111111111111111111	ANGKA
30.	0000000011111100000011110000000011111111	TITIK “. “
31.	0000000011111011000000110000000000001111	TANYA “ ? “
32.	0000000011111100000000000000000000000000	KOMA “ , “
33.	0000001111111111000111110000001111110000	SERU “ ! “
34.	0000111111111111001111110000000000111111	STRIP “ – “

Dari seluruh data pembacaan untuk mendapatkan nilai data input, maka telah didapatkan 5 pola data untuk masing-masing karakter huruf. Sehingga total data keseluruhan dari 34 karakter huruf adalah $34 \times 5 = 170$ pola data. Keseluruhan pola data tersebut nantinya akan dijadikan data input pada proses pembelajaran pada ANN.

3.3.2.2 Proses Pembelajaran Artificial Neural Network

Sebelum dilakukan proses pengenalan huruf braille dengan ANN, maka pada ANN harus dilakukan proses pembelajaran terlebih dahulu. Proses pembelajaran dilakukan dengan memasukkan data input dari huruf a-z dan data input tanda spasi, koma, titik, tanda tanya, tanda angka dan tanda kapital yang telah didapat dari proses pembacaan data input sebelumnya beserta data target yang diinginkan dari masing-masing data pembelajaran tersebut. Proses pembelajaran dilakukan terus-menerus sampai didapat nilai MSE (Mean Square Error) lebih kecil dari nilai error minimum yang telah ditetapkan. Ketika nilai MSE masih lebih besar dari nilai error minimum maka dilakukan proses mundur (backpropagasi) dari layer output ke layer input untuk merubah nilai bobot pada tiap neuron pada masing-masing layer. Kemudian dilakukan proses maju kembali (forward) dan dihitung kembali nilai MSE-nya. Ketika nilai MSE sudah lebih kecil dari nilai error minimum maka proses pembelajaran akan berhenti, dan hasil dari proses pembelajaran tersebut didapatkan nilai-nilai bobot tiap-tiap neuron.

Berikut ini adalah pemilihan nilai parameter yang digunakan dalam proses ANN dan langkah-langkah dalam proses pembelajaran ANN:

- Banyak data = 170 data
- Error minimum = 0,001
- Alpha = 0,75
- Miu = 0,5
- Data input = 40 data
- Node Hidden layer 1 = 80 node
- Node Hidden layer 2 = 50 node
- Node Output layer = 6 node

Berikut adalah langkah-langkah proses pembelajaran dalam ANN:

▪ **Langkah 1 :**

- Inisialisasi Weight (bobot) awal dan bias awal, bobot dan bias menggunakan nilai acak, dengan nilai kecil (antara -0,5 sampai +0,5).
- Tentukan nilai miu (learning rate) dan alpha.
- Tentukan nilai beta (batas minimal error).
- Iterasi=1;

▪ **Langkah2 :**

Selama error belum mencapai nilai yang diinginkan, lakukan langkah-langkah sebagai berikut:

- a. Set aktivasi data input dan target :

Xin[ndata] [1] - Xin[ndata] [40], dan
Target[ndata] [1] - Target[ndata] [40]

- b. Hitung output untuk masing-masing unit pada layer 1:

```
for (m = 1; m <= node_hidden1; m++)
{
    temp=0;
    for (n = 1; n <= banyak_input; n++)
    {
        temp = temp + (WL1[m][n] * XIN[ndata][n]);
    }
    temp = temp + BIAS [1][m];
    Y[1][m] = 1/(1+exp(-alpha * temp));
    Y_G [1][m] = alpha * Y[1][m] * (1-Y[1][m]);
}
```

- c. Hitung output untuk masing-masing unit pada layer 2:

```
for (m = 1; m <= node_hidden2; m++)
{
    temp=0;
    for (n = 1; n <= node_hidden1; n++)
    {
        temp = temp + (WL2[m][n] * Y[1][m]);
    }
    temp = temp + BIAS [2][m];
    Y[2][m] = 1/(1+exp(-alpha * temp));
    Y_G [2][m] = alpha * Y[2][m] * (1-Y[2][m]);
}
```

- d. Hitung output untuk masing-masing unit pada layer Output :

```
for (m = 1; m <= node_output; m++)
{
    temp=0;
    for (n = 1; n <= node_hidden2; n++)
    {
        temp = temp + (WL3[m][n] * Y[2][m]);
    }
    temp = temp + BIAS [3][m];
    Y[3][m] = 1/(1+exp(-alpha * temp));
    Y_G [3][m] = alpha * Y[3][m] * (1-Y[3][m]);
}
```

```
//HITUNG ERROR
    OUT_ERROR [n data][m] = TARGET [ndata][m] - Y[3][m];
    ERROR_AKHIR = ERROR_AKHIR + (powf(OUT_ERROR
[ndata][m],2))/2;
}
```

- e. Update nilai Delta (langkah mundur/backpropagasi):

Delta Layer Output:

```
for (m = 1; m <= node_output; m++)
{
    DELTA [3][m] = OUT_ERROR [ndata][m] * Y_G [3][m];
}
```

Delta Hidden Layer 2:

```
for (m = 1; m <= node_hidden2; m++)
{
    temp = 0;
    for (n = 1; n <= node_output; n++)
    {
        temp = temp + (DELTA[3][n] * WL3[n][m]);
    }
    DELTA [2][m] = temp * Y_G [2][m];
}
```

Delta Hidden Layer 1:

```
for (m = 1; m <= node_hidden1; m++)
{
    temp = 0;
    for (n = 1; n <= node_hidden2; n++)
    {
        temp = temp + (DELTA[2][n] * WL2[n][m]);
    }
    DELTA [1][m] = temp * Y_G [1][m];
}
```

- f. Update nilai Weight (langkah mundur/backpropagasi):

Update bobot layer Output:

```
for (m = 1; m <= node_output; m++)
{
    for (n = 1; n <= node_hidden2; n++)
    {
        WL3[m][n] = WL3[m][n] + (Miu * DELTA [3][m] * Y[2][n]);
    }
}
```

Update bobot Hidden Layer 2:

```
for (m = 1; m <= node_hidden2; m++)
{
    for (n = 1; n<= node_hidden1; n++)
    {
        WL2[m][n] = WL2[m][n] + (Miu * DELTA [2][m] * Y[1][n]);
    }
}
```

Update bobot Hidden Layer 1:

```
for (m = 1; m <= node_hidden1; m++)
{
    for (n = 1; n<= banyak_input; n++)
    {
        WL1[m][n] = WL1[m][n] + (Miu * DELTA [1][m] * Xin[ndata][n]);
    }
}
```

- g. Apakah ndata sudah sama dengan banyak_data, jika belum maka ndata=ndata+1, dan kembali ke **langkah 2 point b**.

Jika ndata sudah sama dengan banyak_data maka iterasi=iterasi+1, dan menuju ke langkah **point h**.

- h. Hitung nilai Mean Square Error (MSE) keseluruhan:

MSE = ERROR_AKHIR / banyak_data;

- i. Jika nilai MSE masih lebih besar dari nilai batas error yang ditentukan, maka ndata=0 dan kembali ke **langkah 2 point b**. Tetapi jika nilai MSE sudah lebih kecil dari nilai batas error yang ditentukan maka proses pembelajaran di hentikan.

Dari proses pembelajaran tersebut sudah didapat nilai bobot optimal tiap neuron dalam setiap layer. Nilai bobot tersebut yang nantinya akan digunakan untuk melakukan perhitungan dari proses ANN dalam melakukan pembacaan huruf braille agar dapat dikenali.

3.3.2.3 Proses Pengenalan Huruf Braille dengan Artificial Neural Network

Setelah didapat nilai bobot optimal tiap neuron pada masing-masing layer dari hasil pembelajaran, maka ANN sudah siap untuk digunakan dalam proses pengenalan huruf braille. Proses pengenalan dilakukan dengan tahapan membaca data input per-satu huruf braille, dimana dalam satu huruf tersebut terdiri dari 40 data input, seperti yang telah dijelaskan pada bab sebelumnya. Dari hasil pembacaan data input pada satu huruf braille, data input tersebut selanjutnya dimasukkan dalam proses ANN. Dalam proses ANN ini hanya dilakukan proses maju (forward) satu kali saja, tanpa proses mundur (backpro) untuk merubah nilai bobot lagi. Dalam satu proses maju tersebut langsung dihasilkan nilai output pada node output yang terdiri dari 6 node, sehingga akan dihasilkan nilai pada Y0 – Y5 sesuai dengan data target yang telah ditentukan dalam proses pembelajaran.

Pada proses ANN digunakan fungsi sigmoid biner, sehingga pada proses ANN akan dihasilkan nilai output yang terletak pada interval 0 sampai 1, maka nilai output pada proses ANN ini berupa nilai pecahan antar 0 sampai 1. Untuk menjadikan nilai tersebut menjadi nilai bulat 0 atau 1, maka perlu ditambahkan nilai threshold untuk menentukan nilai output yang dihasilkan menjadi nilai output bulat 0 atau bulat 1. Aturan penentuannya adalah jika nilai output dari ANN kurang dari nilai threshold maka nilai output tersebut sama dengan 0, sebaliknya jika nilai output dari ANN sama dengan atau lebih besar dari nilai threshold maka nilai output sama dengan 1. Pada penelitian ini ditentukan nilai threshold yang digunakan sebesar 0,75. Dipilih nilai 0,75 dengan asumsi penulis bahwa nilai output dinyatakan benar menjadi 1, jika tingkat kebenaran dari output ANN sebesar 75%. Persamaan (3.8) adalah perhitungan yang digunakan dalam menentukan nilai output berdasarkan nilai threshold yang telah ditentukan.

$$\text{nilai output} = \begin{cases} 0, & \text{jika nilai output} < \text{threshold} \\ 1, & \text{jika nilai output} \geq \text{threshold} \end{cases} \quad (3.8)$$

Persamaan (3.8) tersebut digunakan untuk perhitungan nilai output dari ANN, dari Y0 sampai dengan Y5. Sehingga akan dihasilkan kombinasi nilai biner 6 bit untuk mewakili setiap huruf.

Pada Tabel 3.6 dapat diamati lookup table data keluaran proses ANN untuk setiap huruf.

TABEL 3.6 Data keluaran proses ANN setiap huruf

Data Keluaran Proses ANN Setiap Huruf															
No.	Huruf	Data keluaran ANN						No.	Huruf	Data keluaran ANN					
		Y5	Y4	Y3	Y2	Y1	Y0			Y5	Y4	Y3	Y2	Y1	Y0
0.	SPASI	0	0	0	0	0	0	17.	Q	0	1	0	0	0	1
1.	A	0	0	0	0	0	1	18.	R	0	1	0	0	1	0
2.	B	0	0	0	0	1	0	19.	S	0	1	0	0	1	1
3.	C	0	0	0	0	1	1	20.	T	0	1	0	1	0	0
4.	D	0	0	0	1	0	0	21.	U	0	1	0	1	0	1
5.	E	0	0	0	1	0	1	22.	V	0	1	0	1	1	0
6.	F	0	0	0	1	1	0	23.	W	0	1	0	1	1	1
7.	G	0	0	0	1	1	1	24.	X	0	1	1	0	0	0
8.	H	0	0	1	0	0	0	25.	Y	0	1	1	0	0	1
9.	I	0	0	1	0	0	1	26.	Z	0	1	1	0	1	0
10.	J	0	0	1	0	1	0	27.	Kapital	0	1	1	0	1	1
11.	K	0	0	1	0	1	1	28.	Angka	0	1	1	1	0	0
12.	L	0	0	1	1	0	0	29.	“ . ”	0	1	1	1	0	1
13.	M	0	0	1	1	0	1	30.	“ ? “	0	1	1	1	1	0
14.	N		0	1	1	1	0	31.	“ , ”	0	1	1	1	1	1
15.	O		0	1	1	1	1	32.	“ ! “	1	0	0	0	0	0
16.	P	0	1	0	0	0	0	33.	“ – “	1	0	0	0	0	1

BAB 4

PENGUJIAN DAN ANALISA SISTEM

Pada bab ini akan dibahas mengenai hasil pengujian sistem beserta analisa terhadap terhadap data dari hasil percobaan yang telah dilakukan. Pengujian yang dilakukan meliputi: pengujian pembacaan banyaknya jumlah titik dot hitam pada gambar, pengujian pembacaan koordinat pada tiap titik dot hitam, pengujian penentuan area segmentasi pada tiap area gambar huruf braille, pengujian pengenalan huruf braille untuk bisa diterjemahkan ke teks, pengujian pengenalan huruf braille dengan derajat kemiringan pada gambar yang berbeda-beda, pengujian pengenalan huruf braille dengan derajat kemiringan dari sumber cahaya pada lembar kertas huruf braille, pengujian pengenalan huruf braille dengan gambar dari hasil alat scanner dan yang terakhir pengujian pengenalan huruf braille pada cetakan dua sisi (double side).

4.1 Pengujian Pembacaan Jumlah Titik Dot Hitam

Pengujian pembacaan jumlah titik dot hitam bertujuan untuk mengetahui apakah sistem sudah berjalan dengan baik dalam membaca jumlah titik dot hitam, dan apakah nilai batas yang ditentukan sudah bisa mencakup untuk proses pada 10 data yang diujikan. Jika terjadi kesalahan dalam menghitung banyak titik dot hitam, maka akan mengakibatkan kesalahan pembacaan data koordinat sehingga proses pengenalan huruf braille akan mengalami kesalahan. Cara pengujian proses menghitung banyak titik dot hitam yaitu dengan menentukan nilai batas kumpulan anggota pada area titik dot hitam tersebut. Dari pengujian ini akan dapat ditentukan nilai batas yang digunakan untuk menentukan threshold dalam pembacaan jumlah titik dot hitam pada area gambar braille.

Data hasil pengujian pembacaan jumlah titik dot hitam dapat diamati pada Tabel 4.1. Pada Tabel 4.1 dapat diamati bahwa nilai batas yang dengan nilai 5 merupakan nilai batas dalam proses pembacaan banyaknya titik dot hitam yang hasilnya sesuai dengan jumlah titik dot hitam yang sebenarnya.

TABEL 4.1 Hasil pengujian pembacaan jumlah titik dot hitam

Data	Jumlah titik dot hitam sebenarnya	Jumlah titik dot hitam yang terbaca Dengan pengaturan nilai batas				
		3	5	9	12	15
Data 1	260	261	260	158	120	4
Data 2	306	307	306	194	26	4
Data 3	326	327	326	160	17	2
Data 4	286	287	286	173	12	1
Data 5	320	320	320	205	26	8
Data 6	313	313	313	155	15	7
Data 7	333	334	333	202	21	3
Data 8	333	334	333	220	36	7
Data 9	293	294	293	181	15	4
Data 10	253	254	253	145	15	2

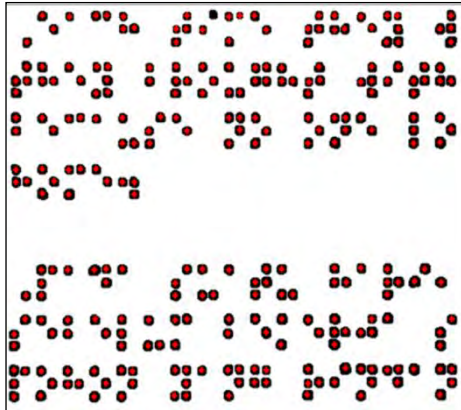
Dari data pengujian pada Tabel 4.1 diatas maka pada penelitian ini ditentukan nilai batas threshold dalam pembacaan jumlah titik dot hitam adalah bernilai 5.

4.2 Pengujian Pembacaan Koordinat Pada Setiap Titik Dot Hitam

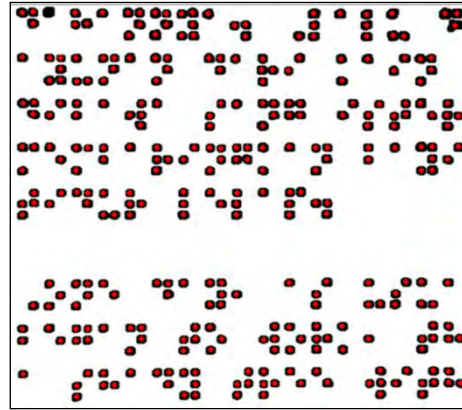
Pengujian selanjutnya yaitu pengujian pembacaan koordinat pada setiap titik dot hitam. Pengujian ini dilakukan untuk mengetahui apakah sistem sudah bisa membaca titik koordinat pada setiap titik dot hitam dengan benar. Pengujian dilakukan dengan menggunakan 10 gambar yang berbeda. Dari hasil pengujian ini akan didapatkan nilai titik koordinat dari masing-masing titik dot hitam, nilai tersebut selanjutnya akan dijadikan untuk proses selanjutnya yaitu proses mencari nilai koordinat rata-rata pada baris dan kolom serta proses segmentasi.

Hasil pembacaan koordinat pada tiap titik dot ditandai dengan titik warna merah pada area tengah dari titik dot hitam seperti terlihat pada Gambar 4.1.

Pada Gambar 4.1 (a – j) merupakan hasil pengujian pembacaan koordinat tiap titik dot hitam.



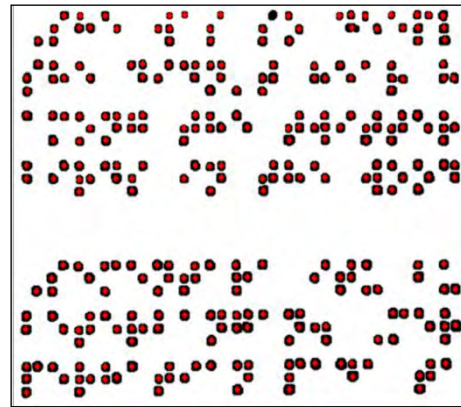
(b) Data 1



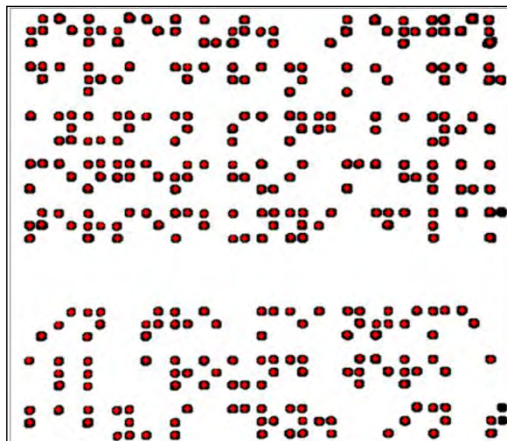
(a) Data 2



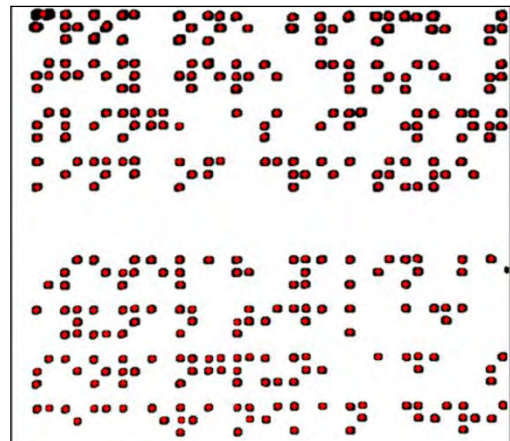
(c) Data 3



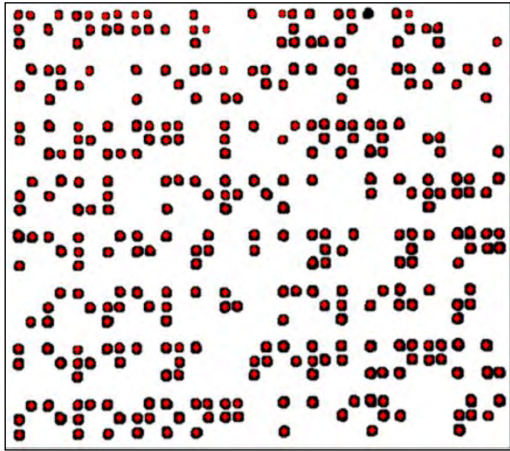
(d) Data 4



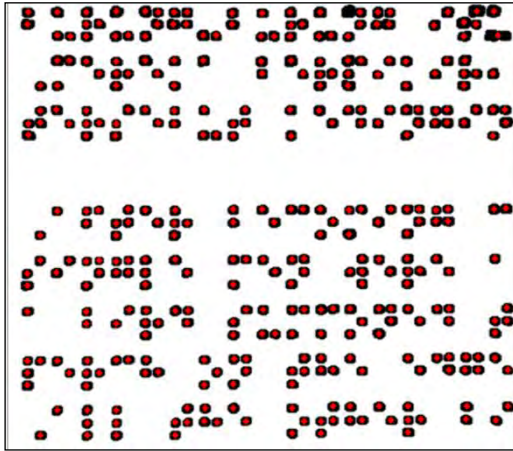
(e) Data 5



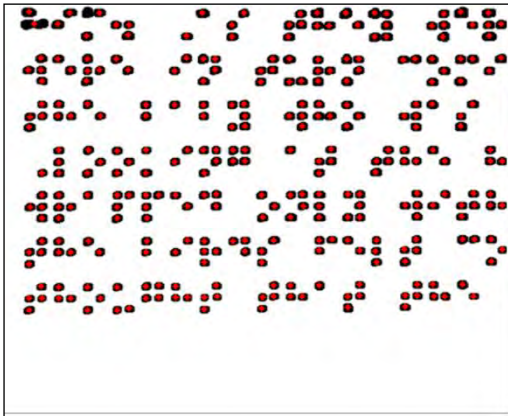
(f) Data 6



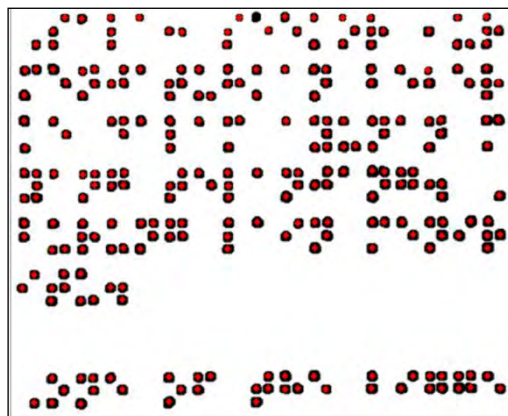
(g) Data 7



(h) Data 8



(j) Data 9

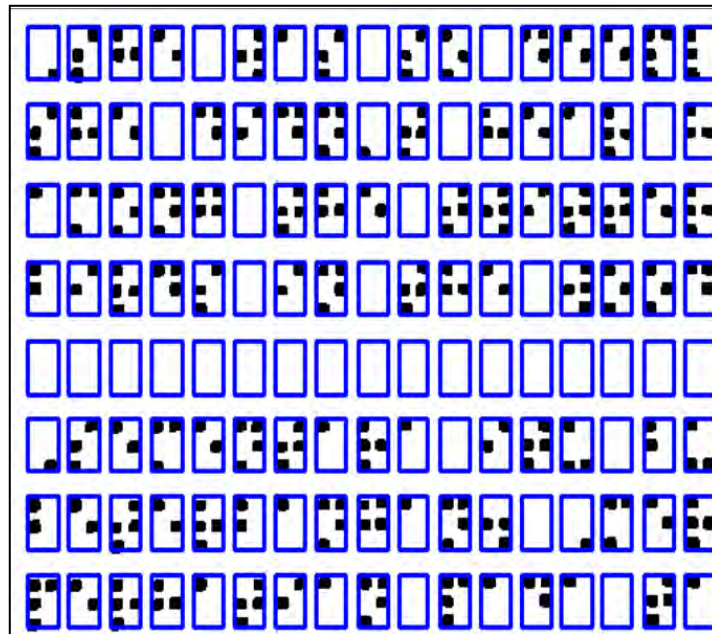


(i) Data 10

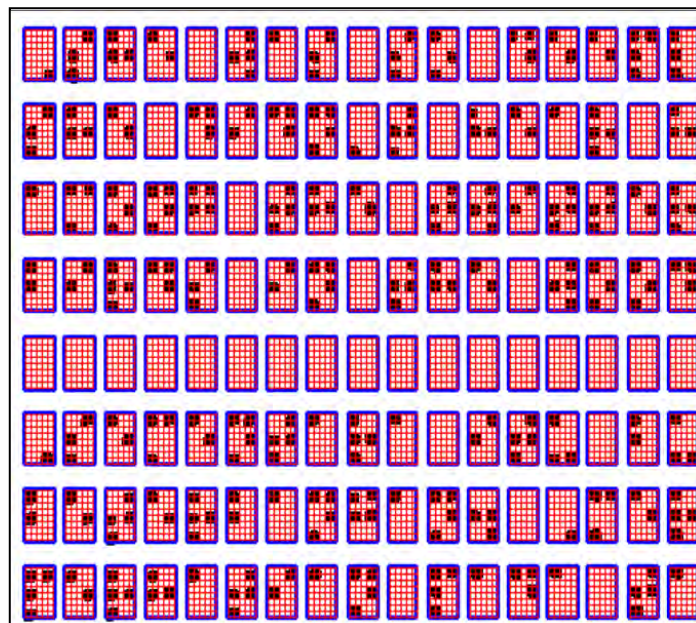
Gambar 4.1 (a – j) Hasil pengujian pembacaan koordinat tiap titik dot hitam.

4.3 Pengujian Penentuan Area Segmentasi Pada Setiap Huruf Braille

Pengujian penentuan area segmentasi pada tiap huruf braille ini bertujuan untuk mengamati apakah area segmentasi yang dilakukan sudah sesuai dengan area tiap huruf braille. Pada Gambar 4.2 terlihat bahwa pada area gambar braille keseluruhan dilakukan segmentasi tiap satu huruf braille. Kemudian setiap satu huruf braille disegmentasi lagi menjadi area yang kecil-kecil 5 kolom dan 8 baris sehingga total terdapat 40 area segmentasi kecil pada area satu huruf. Untuk segmentasi kecil dapat diamati pada Gambar 4.3.



Gambar 4.2 Penentuan area segmentasi pada tiap huruf braille



Gambar 4.3 Segmentasi pada tiap huruf braille dijadikan 40 area segmentasi

Dari hasil pengujian dan pengamatan Gambar 4.2 dan Gambar 4.3 maka sistem sudah mampu menentukan area segmentasi setiap huruf braille dengan hasil yang tepat dan benar.

4.4 Pengujian Pengenalan Huruf Braille di Terjemahkan ke Teks

Pada pengujian ini adalah pengujian kemampuan sistem ini dalam pengenalan huruf braille untuk diterjemahkan ke teks. Pengujian dilakukan menggunakan berbagai cetakan kertas braille yang bervariasi, antara lain:

1. Cetakan kertas huruf braille tanpa spasi antar huruf
2. Cetakan kertas huruf braille dengan 1 (satu) spasi antar huruf.
3. Cetakan kertas huruf braille dengan 2 (dua) spasi antar huruf.
4. Cetakan kertas huruf braille dengan 3 (tiga) spasi antar huruf.
5. Cetakan buku braille dari buku cerita “Hua Lo Puu” sebanyak 10 halaman.
6. Cetakan buku braille hasil dari cetakan printer Braillo-400 (di Lab. Elektro-ITS) dengan cerita Si Kancil dan Si Siput sebanyak 5 halaman.

4.4.1 Pengujian Pengenalan Huruf Braille Pada Cetakan Tanpa Spasi

Pada pengujian ini sistem diuji coba untuk mengenali hasil cetakan huruf braille cetakan dari huruf A-Z, angka 0-9, tanda titik, tanda, koma, tanda tanya, spasi dan tanda angka. Dimana pada cetakan tersebut jarak antar huruf braille tidak diberikan spasi. Pengujian ini dilakukan untuk mengetahui tingkat akurasi sistem dalam mengenali cetakan huruf braille tanpa ada jarak spasi. Pada Gambar 4.4 merupakan gambar cetakan huruf braille tanpa jarak spasi.



Gambar 4.4 Cetakan huruf braille tanpa jarak spasi antar huruf

TABEL 4.2 Data pengujian pengenalan huruf braille pada cetakan tanpa spasi

Percobaan	Huruf sebenarnya	Terbaca	Tidak terbaca	Kesalahan (%)	Akurasi (%)
1	910	905	5	0,55 %	99,45 %
2	910	901	9	0,99 %	99,01 %
3	910	903	7	0,77 %	99,23 %
4	910	903	7	0,77 %	99,23 %
5	910	902	8	0,88 %	99,12 %
6	910	904	6	0,66 %	99,34 %
7	910	902	8	0,88 %	99,12 %
8	910	903	7	0,77 %	99,23 %
9	910	904	6	0,66 %	99,34 %
10	910	903	7	0,77 %	99,23 %
Rata – rata tingkat akurasi					99,23 %

Dari hasil data pengujian pada Tabel 4.2 dapat diketahui bahwa untuk mengenali cetakan huruf braille tanpa ada jarak spasi antar huruf braille, sistem memiliki tingkat akurasi sebesar 99,23%.

4.4.2 Pengujian Pengenalan Huruf Braille Pada Cetakan Jarak Satu Spasi

Pada pengujian ini sistem diuji coba untuk mengenali hasil cetakan huruf braille cetakan dari huruf A-Z, angka 0-9, tanda titik, tanda koma, tanda tanya, spasi dan tanda angka. Dimana pada cetakan tersebut jarak antar huruf braille diberikan jarak satu spasi. Pengujian ini dilakukan untuk mengetahui tingkat akurasi sistem dalam mengenali cetakan huruf braille dengan ada jarak satu spasi.



Gambar 4.5 Cetakan huruf braille dengan jarak satu spasi antar huruf

Pada Gambar 4.5 merupakan gambar cetakan huruf braille dengan jarak satu spasi dan pada Tabel 4.3 merupakan data hasil pengujian pengenalan huruf braille pada cetakan dengan jarak antar huruf satu spasi.

TABEL 4.3 Data pengujian pengenalan huruf braille pada cetakan dengan jarak satu spasi

Percobaan	Huruf sebenarnya	Terbaca	Tidak terbaca	Kesalahan (%)	Akurasi (%)
1	910	908	2	0,22 %	99,78 %
2	910	907	3	0,33 %	99,67 %
3	910	907	3	0,33 %	99,67 %
4	910	907	3	0,33 %	99,67 %
5	910	908	2	0,22 %	99,78 %
6	910	906	4	0,44 %	99,56 %
7	910	907	3	0,33 %	99,67 %
8	910	907	3	0,33 %	99,67 %
9	910	907	3	0,33 %	99,67 %
10	910	907	3	0,33v	99,67 %
Rata – rata tingkat akurasi					99,68 %

Dari hasil data pengujian pada Tabel 4.3 dapat diketahui bahwa untuk mengenali cetakan huruf braille dengan jarak satu spasi antar huruf braille, sistem memiliki tingkat akurasi sebesar 99,68%.

4.4.3 Pengujian Pengenalan Huruf Braille Pada Cetakan Jarak Dua Spasi

Pada pengujian ini sistem diuji coba untuk mengenali hasil cetakan huruf braille cetakan dari huruf A-Z, angka 0-9, tanda titik, tanda, koma, tanda tanya, spasi dan tanda angka. Dimana pada cetakan tersebut jarak antar huruf braille diberikan jarak dua spasi. Pengujian ini dilakukan untuk mengetahui tingkat akurasi sistem dalam mengenali cetakan huruf braille dengan ada jarak antar huruf braille dua spasi.

Pada Gambar 4.6 merupakan gambar cetakan huruf braille dengan jarak dua spasi dan pada Tabel 4.4 merupakan data hasil pengujian pengenalan huruf braille pada cetakan dengan jarak antar huruf dua spasi.



Gambar 4.6 Cetakan huruf braille dengan jarak dua spasi antar huruf

Pada pengujian pengenalan huruf braille pada cetakan dua spasi ini didapat data hasil pengujian seperti terlihat pada Tabel 4.4. Dari hasil data pengujian pada Tabel 4.4 dapat diketahui bahwa untuk mengenali cetakan huruf braille dengan jarak dua spasi antar huruf braille, sistem memiliki tingkat akurasi sebesar 99,84%.

TABEL 4.4 Data pengujian pengenalan huruf braille pada cetakan dengan jarak dua spasi

Percobaan	Huruf sebenarnya	Terbaca	Tidak terbaca	Kesalahan (%)	Akurasi (%)
1	910	909	1	0,11 %	99,89 %
2	910	908	2	0,22 %	99,78 %
3	910	908	2	0,22 %	99,78 %
4	910	908	2	0,22 %	99,78 %
5	910	909	1	0,11 %	99,89 %
6	910	907	3	0,33 %	99,67 %
7	910	909	1	0,11 %	99,89 %
8	910	909	1	0,11 %	99,89 %
9	910	909	1	0,11 %	99,89 %
10	910	909	1	0,11 %	99,89 %
Rata – rata tingkat akurasi					99,84 %

4.4.4 Pengujian Pengenalan Huruf Braille Pada Cetakan Jarak Tiga Spasi

Pada pengujian ini sistem diuji coba untuk mengenali hasil cetakan huruf braille cetakan dari huruf A-Z, angka 0-9, tanda titik, tanda koma, tanda tanya, spasi dan tanda angka. Dimana pada cetakan tersebut jarak antar huruf braille diberikan jarak tiga spasi. Pengujian ini dilakukan untuk mengetahui tingkat akurasi sistem dalam mengenali cetakan huruf braille dengan ada jarak antar huruf braille tiga spasi.

TABEL 4.5 Data pengujian pengenalan huruf braille pada cetakan dengan jarak tiga spasi

Percobaan	Huruf sebenarnya	Terbaca	Tidak terbaca	Kesalahan (%)	Akurasi (%)
1	910	876	34	3,74	96,26
2	910	876	34	3,74	96,26
3	910	875	35	3,85	96,15
4	910	876	34	3,74	96,26
5	910	876	34	3,74	96,26
6	910	875	35	3,85	96,15
7	910	876	34	3,74	96,26
8	910	876	34	3,74	96,26
9	910	875	35	3,85	96,15
10	910	876	34	3,74	96,26
Rata – rata tingkat akurasi					96,23



Gambar 4.7 Cetakan huruf braille dengan jarak tiga spasi antar huruf

Dari hasil data pengujian pada Tabel 4.5 dapat diketahui bahwa untuk mengenali cetakan huruf braille dengan jarak tiga spasi antar huruf braille, tingkat akurasi sistem dalam proses pengenalan huruf braille menurun menjadi 96,23%.

4.4.5 Pengujian Pengenalan Huruf Braille Pada Cetakan Buku Cerita

Pengujian pengenalan huruf braille selanjutnya dilakukan langsung pada buku cetakan huruf braille yang terdiri atas 10 halaman lembar kertas braille. Buku huruf braille yang digunakan untuk percobaan merupakan buku terbitan dari Direktorat Pembinaan Sekolah Luar Biasa Ditjen Mandikdasmen, Depdiknas. Buku cerita tersebut berjudul “Hua Lo Puu”. Pengujian ini dilakukan untuk mengetahui tingkat akurasi sistem jika digunakan pada buku cetakan huruf braille hasil terbitan dari pemerintah.

Pada Tabel 4.6 adalah data hasil pengujian pengenalan huruf braille pada buku cerita cetakan huruf braille yang berjudul “Hua Lo Puu”.

TABEL 4.6 Data pengujian pengenalan huruf braille pada cetakan buku cerita yang berjudul “Hua Lo Puu”.

Halaman	Huruf sebenarnya	Terbaca	Tidak terbaca	Kesalahan (%)	Akurasi (%)
1	910	898	0	0 %	100 %
2	910	901	0	0 %	100 %
3	910	890	1	0,11 %	99,89 %
4	910	903	4	0,44 %	99,56 %
5	910	831	0	0 %	100 %
6	910	898	0	0 %	100 %
7	910	892	1	0,11 %	99,89 %
8	910	889	3	0,34 %	99,66 %
9	910	871	3	0,34 %	99,66 %
10	910	882	2	0,23 %	99,77 %
Rata – rata tingkat akurasi					99,84 %

Dari hasil data pengujian pada Tabel 4.6 dapat diketahui bahwa untuk mengenali cetakan huruf braille pada cetakan buku cerita yang berjudul “Hua Lo Puu”, tingkat akurasi sistem dalam proses pengenalan huruf braille sebesar 99,84%.

4.4.6 Pengujian Pengenalan Huruf Braille Hasil Cetakan Printer Braillo-400 (di Lab. Elektro ITS)

Pengujian pengenalan huruf braille selanjutnya juga dilakukan pada buku cetakan huruf braille hasil cetakan dari printer braille merk Braillo-400 yang ada di Lab. AJ-304 Teknik Elektro ITS. Buku cetakan tersebut merupakan buku cerita yang berjudul “Si Kancil dan Si Siput”, cetakan terdiri atas 5 halaman lembar kertas braille. Pengujian ini dilakukan untuk mengetahui tingkat akurasi sistem jika digunakan pada buku cetakan huruf braille hasil cetakan dari printer Braillo-400 yang ada di Lab. AJ-304 Teknik Elektro ITS.

Pada Tabel 4.7 adalah data hasil pengujian pengenalan huruf braille pada buku cetakan huruf braille hasil cetakan dari printer Braillo-400 yang ada di Lab. AJ-304 Teknik Elektro ITS.

TABEL 4.7 Data pengujian pengenalan huruf braille pada buku cetakan huruf braille hasil cetakan dari printer Braillo-400 yang ada di Lab. AJ-304 Teknik Elektro ITS.

Halaman	Huruf sebenarnya	Terbaca	Tidak terbaca	Kesalahan (%)	Akurasi (%)
1	873	873	0	0 %	100 %
2	891	888	3	0,34 %	99,66 %
3	879	876	3	0,34 %	99,66 %
4	883	875	8	0,91 %	99,09 %
5	293	292	1	0,34v	99,66 %
Rata – rata tingkat akurasi					99,61 %

Dari hasil data pengujian pada Tabel 4.7 dapat diketahui bahwa untuk mengenali cetakan huruf braille pada cetakan huruf braille hasil cetakan dari printer Braillo-400 yang ada di Lab. AJ-304 Teknik Elektro ITS, tingkat akurasi sistem dalam proses pengenalan huruf braille sebesar 99,61%.

4.5 Pengujian Pengenalan Huruf Braille Dengan Tingkat Kemiringan Pada Gambar Dari 0 Derajat Sampai 1,5 Derajat.

Pengujian selanjutnya adalah pengujian kemampuan sistem saat gambar yang diproses diposisikan miring dengan sudut kemiringan yang bervariasi. Pengujian dilakukan menggunakan 10 data gambar huruf braille yang berbeda dan sudut kemiringan dari gambar dibuat bervariasi dari -1,5 derajat sampai 1,5 derajat kemiringan.

TABEL 4.8 Data pengujian pengenalan huruf braille dengan tingkat kemiringan pada gambar dari 0 derajat sampai 1,5 derajat.

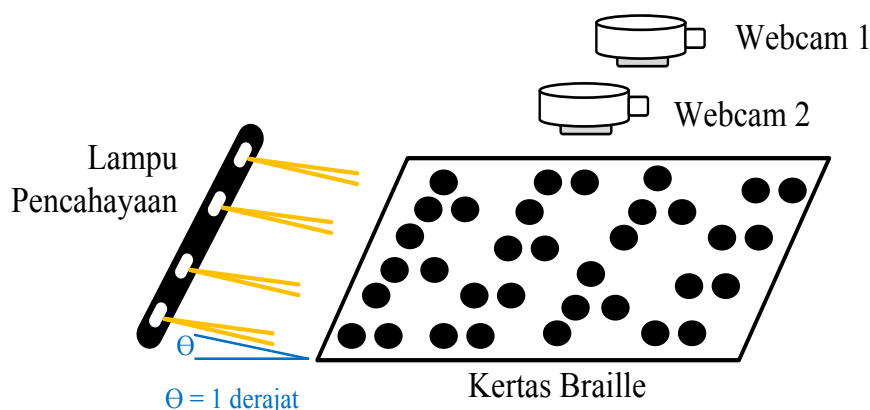
Derajat Kemiringan	Semua Data 10 Gambar				
	Huruf sebenarnya	Terbaca	Tidak terbaca	Kesalahan (%)	Akurasi (%)
-1,5	1229	9	1220	99,27 %	0,73 %
-1,25	1229	665	564	45,89 %	54,11 %
-1	1229	1220	9	0,74 %	99,26 %
-0,75	1229	1217	12	0,98 %	99,02 %
-0,5	1229	1225	4	0,33 %	99,67 %
-0,25	1229	1224	5	0,41 %	99,59 %
0°	1229	1226	3	0,25 %	99,75 %
0,25°	1229	1225	4	0,33 %	99,67 %
0,5°	1229	1225	4	0,33 %	99,67 %
0,75°	1229	1214	15	1,23 %	98,77 %
1°	1229	1221	8	0,66 %	99,34 %
1,25°	1229	731	498	40,53 %	59,47 %
1,5	1229	24	1205	98,05 %	1,95 %

Dari hasil percobaan yang diperoleh tingkat akurasi sistem pengenalan huruf braille pada gambar dengan kemiringan 0 derajat bisa mencapai keakuratan 99%. Tingkat akurasi sistem mulai menurun saat derajat kemiringan gambar mulai 1,25 derajat atau -1,25 derajat, dan sistem tidak mampu mengenali sama sekali saat derajat kemiringan gambar pada 1,5 derajat atau -1,5 derajat.

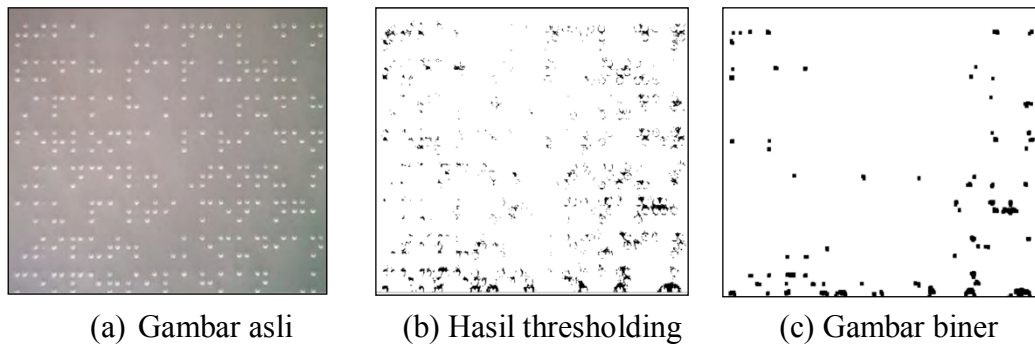
4.6 Pengujian Pengenalan Huruf Braille Dengan Tingkat Kemiringan Sumber Cahaya Terhadap Kertas Braille.

Pengujian selanjutnya adalah pengujian kemampuan sistem ini dalam pengenalan huruf braille untuk diterjemahkan ke teks dengan merubah-ubah derajat kemiringan sumber cahaya yang mengarah ke kertas braille. Pengujian dilakukan dengan menempatkan sumber cahaya pada sisi samping kertas braille dengan jarak ± 25 cm dan dengan derajat kemiringan sumber cahaya yang mengarah ke kertas braille mulai dari 1° , 5° , 10° , 15° , 20° , 25° , 30° , 45° , 60° dan 90° .

Tujuan dari pengujian ini adalah untuk mengetahui tingkat akurasi sistem dalam melakukan proses pengenalan huruf braille pada gambar dengan posisi kemiringan sumber cahaya yang berbeda dan untuk mengetahui posisi besar derajat kemiringan sumber cahaya yang tepat agar proses pengenalan huruf braille dapat berhasil dengan baik. Pada Gambar 4.8 merupakan hasil pengambilan gambar menggunakan webcam kamera, dengan sudut kemiringan sumber cahaya sebesar 1 derajat. Terlihat pada gambar asli yang dihasilkan, pada area dekat titik timbul tidak membentuk bayangan sehingga saat diproses thresholding gambar yang dihasilkan tidak beraturan dan pada Gambar biner terlihat titik dot hitam yang nampak lebih sedikit dari Gambar aslinya. Jadi dapat diambil kesimpulan sumber cahaya dengan sudut kemiringan 1 derajat hasilnya tidak baik.

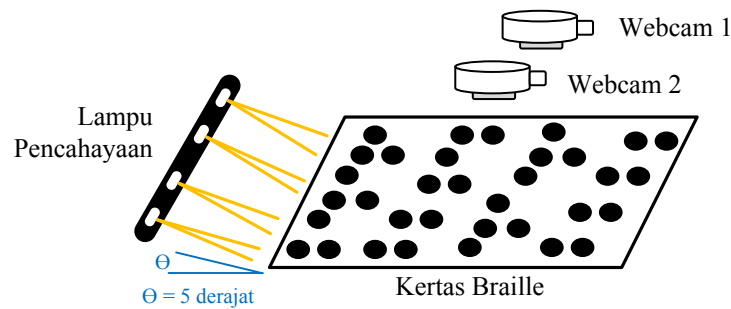


Gambar 4.8 Ilustrasi penempatan sumber pencahayaan dengan kemiringan 1 derajat (1°) terhadap kertas braille

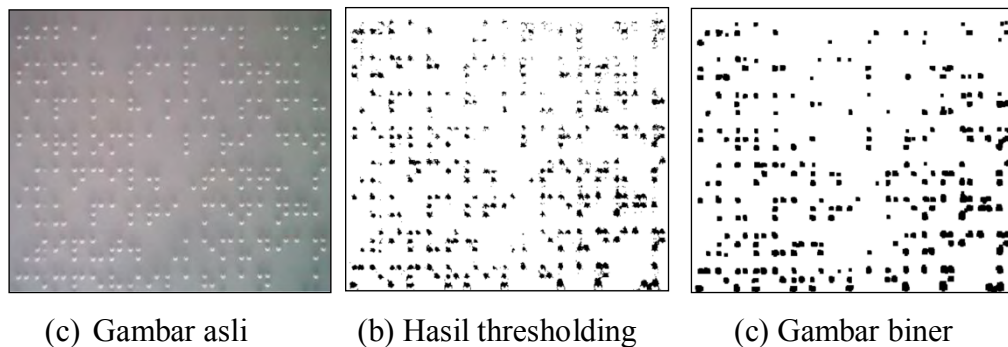


Gambar 4.9 Gambar hasil sumber pencahayaan kemiringan 1 derajat (1°)

Selanjutnya dicoba penempatan sumber pencahayaan dengan kemiringan 5 derajat (5°). Pada Gambar 4.11 terlihat pada gambar asli yang dihasilkan, pada area dekat titik timbul sudah membentuk bayangan namun belum begitu jelas sehingga saat diproses thresholding gambar yang dihasilkan masih tidak beraturan dan pada Gambar biner terlihat titik dot hitam yang nampak lebih sedikit dari Gambar aslinya. Jadi dapat diambil kesimpulan sumber cahaya dengan sudut kemiringan 5 derajat hasilnya masih belum baik.

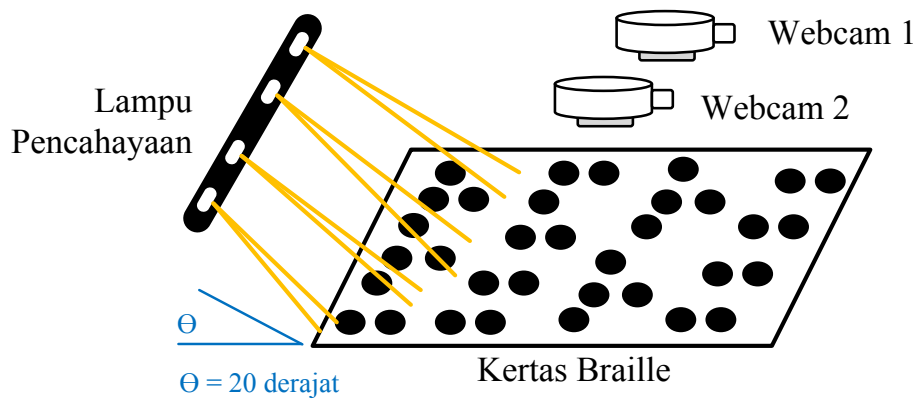


Gambar 4.10 Ilustrasi penempatan sumber pencahayaan dengan kemiringan 5 derajat (5°) terhadap kertas braille

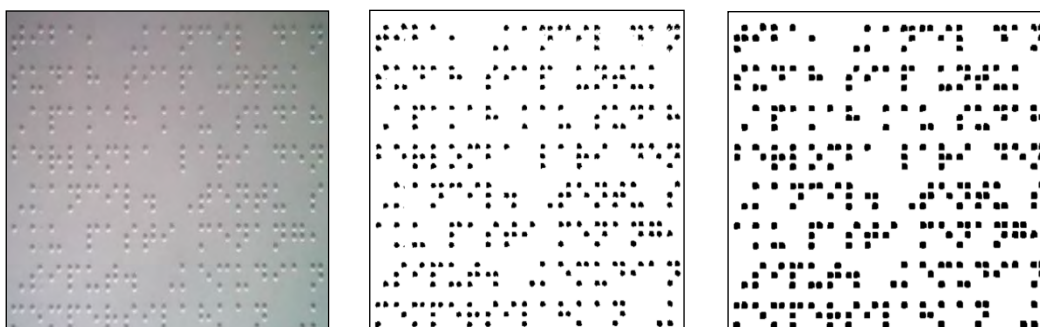


Gambar 4.11 Gambar hasil sumber pencahayaan kemiringan 5 derajat (5°)

Selanjutnya sudut kemiringan penempatan sumber pencahayaan dinaikkan lagi menjadi 20 derajat. Pada Gambar 4.12 terlihat ilustrasi penempatan sumber cahaya dengan kemiringan 20 derajat terhadap kertas braille, dan pada Gambar 4.13 merupakan gambar asli yang dihasilkan, pada area dekat titik timbul sudah membentuk bayangan dengan jelas sehingga saat diproses thresholding gambar yang dihasilkan terlihat titik-titik dot hitam dengan jelas. Area titik hitam tampak dengan jelas dan jumlah titik hitam sesuai dengan aslinya. Jadi dapat diambil kesimpulan sumber cahaya dengan sudut kemiringan 20 derajat hasilnya baik.



Gambar 4.12 Ilustrasi penempatan sumber pencahayaan dengan kemiringan 20 derajat (20°) terhadap kertas braille



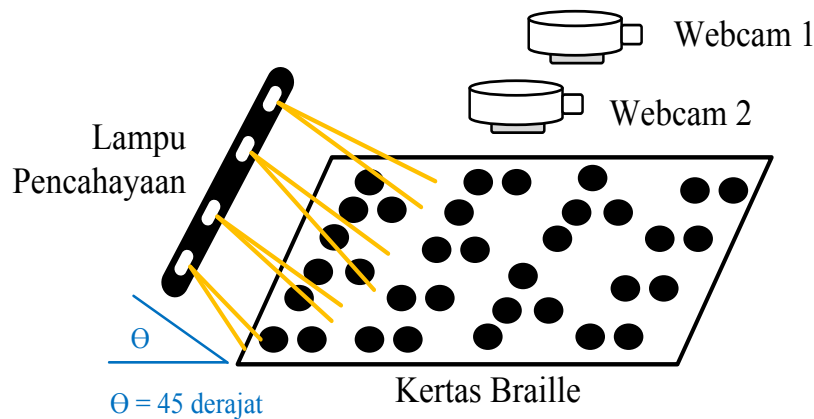
(d) Gambar asli

(b) Hasil thresholding

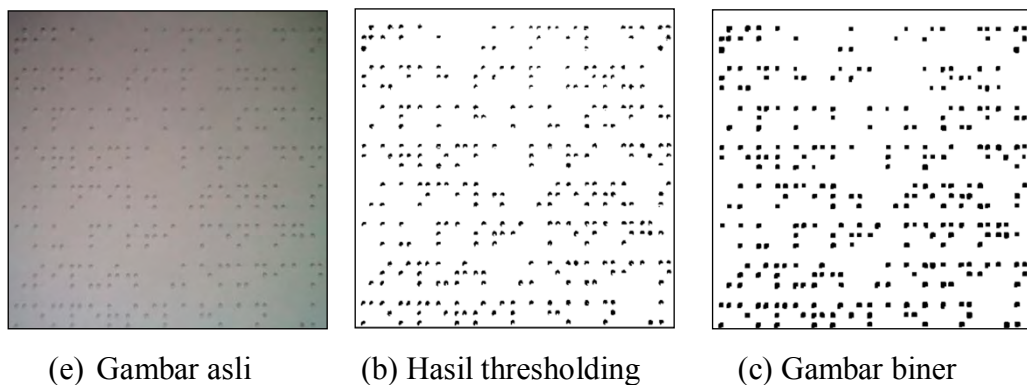
(c) Gambar biner

Gambar 4.13 Gambar hasil sumber pencahayaan kemiringan 20 derajat (20°)

Pengujian dilakukan lagi dengan menaikkan lagi sudut pencahayaan sumber cahaya terhadap kertas braille menjadi 45 derajat. Pada Gambar 4.14 terlihat ilustrasi penempatan sumber cahaya dengan kemiringan 45 derajat terhadap kertas braille, dan pada Gambar 4.15 merupakan gambar asli yang dihasilkan, gambar hasil proses thresholding dengan gambar sumber dari penempatan sumber cahaya dengan kemiringan 45 derajat pada bagian atas terlihat kurang begitu jelas. Bagian atas gambar terlihat titik dot hitam lebih kecil jika dibandingkan dengan area bagian bawah, sehingga terlihat tidak merata. Jadi dapat diambil kesimpulan sumber cahaya dengan sudut kemiringan 45 derajat hasilnya kurang baik.

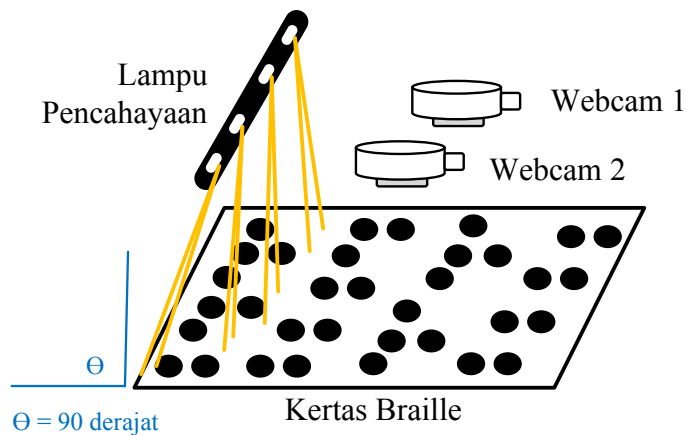


Gambar 4.14 Ilustrasi penempatan sumber pencahayaan dengan kemiringan 45 derajat (45°) terhadap kertas braille

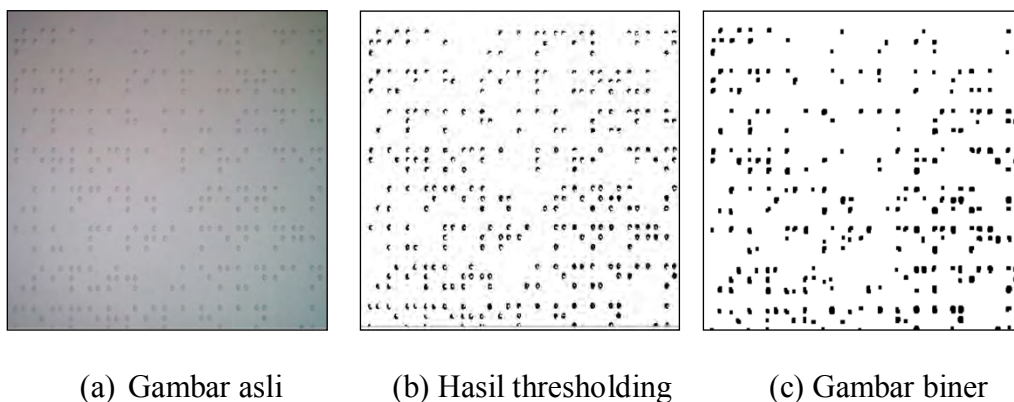


Gambar 4.15 Gambar hasil sumber pencahayaan kemiringan 45 derajat (45°)

Pengujian dilakukan lagi dengan menaikkan lagi sudut pencahayaan sumber cahaya terhadap kertas braille menjadi 90 derajat sehingga menjadi tegak lurus terhadap kertas braille. Pada Gambar 4.16 terlihat ilustrasi penempatan sumber cahaya dengan kemiringan 90 derajat terhadap kertas braille, dan pada Gambar 4.17 merupakan gambar asli yang dihasilkan, gambar hasil proses thresholding dengan gambar sumber dari penempatan sumber cahaya dengan kemiringan 90 derajat pada bagian atas semakin terlihat kurang begitu jelas. Bagian atas gambar terlihat titik dot hitam lebih kecil jika dibandingkan dengan area bagian bawah, sehingga terlihat tidak merata. Jadi dapat diambil kesimpulan sumber cahaya dengan sudut kemiringan 90 derajat hasilnya semakin kurang baik.



Gambar 4.16 Ilustrasi penempatan sumber pencahayaan dengan kemiringan 90 derajat (90°) terhadap kertas braille



Gambar 4.17 Gambar hasil sumber pencahayaan kemiringan 90 derajat (90°)

Dari serangkaian pengujian akurasi sistem terhadap sudut pencahayaan sumber cahaya yang diarahkan ke kertas braille mulai dari sudut 1°, 5°, 10°, 15°, 20°, 25°, 30°, 45°, 60° dan 90°. Telah didapatkan hasil data pengujian seperti yang tertera pada Tabel 4.9.

TABEL 4.9 Data pengujian pengenalan huruf braille dengan tingkat kemiringan sumber cahaya pada gambar.

Sudut Cahaya	Huruf sebenarnya	Terbaca	Tidak terbaca	Kesalahan (%)	Akurasi (%)
1°	129	4	125	96,90 %	3,10 %
5°	129	64	65	50,39 %	49,61 %
10°	129	114	15	11,63 %	88,37 %
15°	129	126	3	2,33 %	97,67 %
20°	129	129	0	0,00 %	100,00 %
25°	129	129	0	0,00 %	100,00 %
30°	129	127	2	1,55 %	98,45 %
45°	129	115	14	10,85 %	89,15 %
60°	129	102	27	20,93 %	79,07 %
90°	129	74	55	42,64 %	57,36 %

Dari data hasil pengujian penempatan sumber cahaya terhadap lembar kertas braille dengan posisi sudut pencahayaan yang berbeda. Dapat diamati data hasil pengujian pada Tabel 4.9 bahwa sudut pencahayaan antara 20 derajat sampai dengan 25 derajat menghasilkan hasil pemrosesan pengenalan huruf braille dengan akurasi 100%, sedangkan sudut pencahayaan yang lain berakibat tingkat akurasi pengenalan huruf braille menjadi turun. Jadi dapat disimpulkan sudut pencahayaan yang baik pada sistem ini adalah 20-25 derajat.

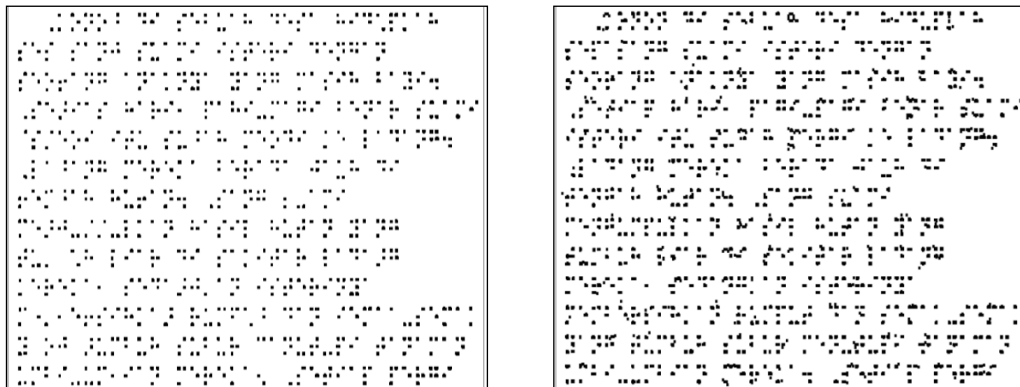
4.7 Pengujian Sistem Pengenalan Huruf Braille Dengan Pengambilan Gambar Menggunakan Alat Scanner

Pada pengujian ini dilakukan untuk menguji sistem untuk melakukan pengenalan huruf braille pada gambar braille hasil dari alat scanner. Pengambilan gambar dilakukan dengan cara men-scan lembar kertas huruf braille cetakan 1 sisi menggunakan alat scanner. Alat scanner yang digunakan merk Epson tipe WorkForce WF7511. Pada saat proses scanner digunakan pemilihan resolusi sebesar 75 dpi. Gambar hasil pengambilan gambar cetakan huruf braille pada cetakan 1 sisi seperti terlihat pada Gambar 4.18.



Gambar 4.18. Hasil pengambilan gambar braille cetakan 1 sisi dengan alat scanner

Pada proses thresholding dicoba dengan menggunakan General Threshold dan Adaptive Threshold. Proses dengan General Threshold didapatkan hasil yang lebih baik, hasil proses thresholding seperti terlihat pada Gambar 4.19. Pada proses General Threshold digunakan nilai threshold 220.



Gambar 4.19. (a). Hasil general threshold, (b) Hasil adaptive threshold

Pada Gambar 4.19 dapat diamati untuk proses thresholding sumber gambar dari hasil alat scanner menggunakan general threshold hasilnya lebih baik jika dibandingkan dengan menggunakan adaptive threshold. Untuk hasil dengan general threshold hasilnya lebih jelas, sedangkan hasil dengan adaptive threshold hasilnya cenderung berhimpitan antar titik dot, dan hal tersebut akan membuat kesalahan saat proses pendeteksian kontur area titik dot, karena bisa saja dua area titik dot dianggap menjadi satu titik dot. Berikut adalah data hasil pengujian sistem untuk pengenalan huruf braille gambar dari hasil scanner.

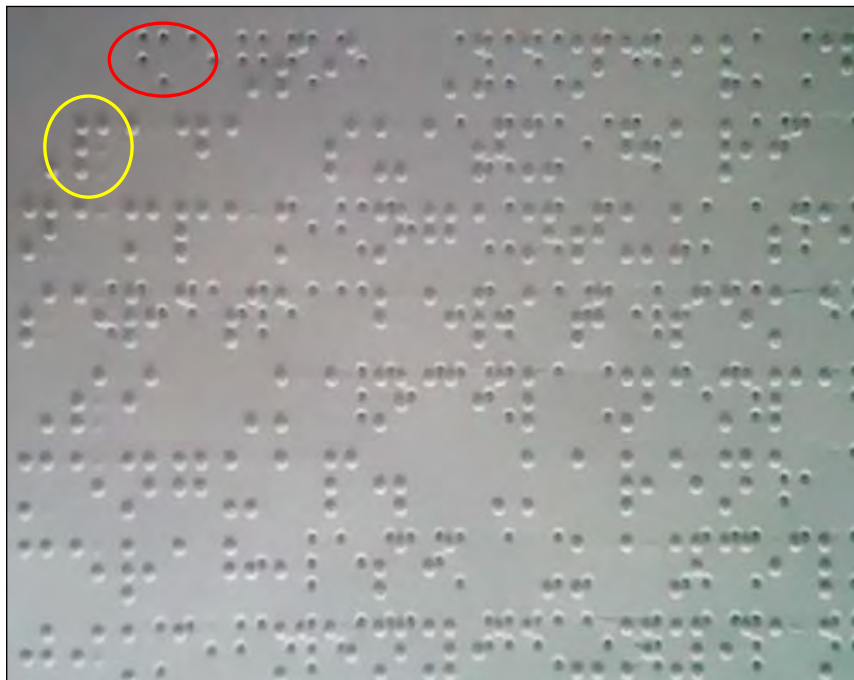
TABEL 4.10. Data pengujian pengenalan sistem untuk pengenalan huruf braille gambar dari hasil scanner

Percobaan	Huruf sebenarnya	Terbaca	Tidak terbaca	Kesalahan (%)	Akurasi (%)
1	443	423	20	3,74 %	95,4 %
2	443	423	20	3,74 %	95,4 %
3	443	424	19	3,85 %	95,7 %
4	443	423	20	3,74 %	95,4 %
5	443	423	20	3,74 %	95,4 %
6	443	423	20	3,85 %	95,4 %
7	443	423	20	3,74 %	95,4 %
8	443	423	20	3,74 %	95,4 %
9	443	423	20	3,85 %	95,4 %
10	443	423	20	3,74 %	95,4 %
Rata – rata tingkat akurasi					95,43 %

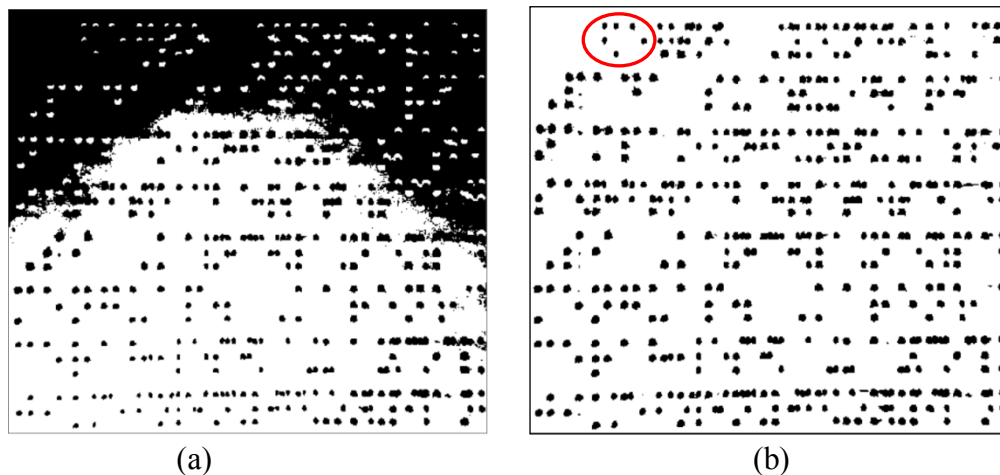
Dengan mengamati hasil pengujian pada Tabel 4.10 maka dapat disimpulkan sistem dapat melakukan pengenalan pada gambar hasil scanner menggunakan resolusi 75 dpi dengan tingkat akurasi sebesar 95,43%.

4.8 Pengujian Sistem Pengenalan Huruf Braille Pada Cetakan Double Side (2 Sisi) Pengambilan Gambar dengan Kamera Webcam

Telah dilakukan pengujian pada cetakan double side, namun sistem tidak bisa mengenali setiap karakter dengan benar. Hal tersebut dikarenakan bagian belakang dari titik timbul yang mengarah ke sisi 2 masih terbaca, sehingga kondisi tersebut dibaca oleh sistem sebagai titik timbul yang mengarah ke sisi 1 juga, seperti terlihat pada Gambar 4.20. Pada Gambar 4.20 dapat diamati bahwa tanda garis lingkaran warna merah merupakan bagian belakang dari titik timbul pada sisi ke 2. Sedangkan pada tanda garis lingkaran warna kuning merupakan titik timbul pada sisi 1.



Gambar 4.20 Hasil pengambilan gambar huruf braille pada cetakan 2 sisi

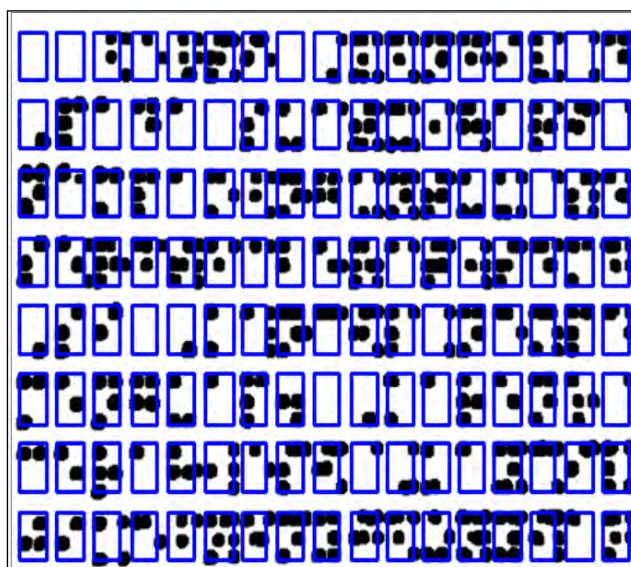


Gambar 4.21 (a) Hasil proses thresholding dengan General Threshold
(b) Hasil proses thresholding dengan Adaptive Threshold

Pada proses thresholding dicoba dengan menggunakan General Threshold dan Adaptive Threshold. Proses dengan Adaptive Threshold didapatkan hasil yang lebih baik, namun titik belakang dari sisi ke 2 masih tetap terbaca seperti terlihat pada Gambar 4.21(b) dan ditandai dengan garis lingkaran warna merah. Sudah dicoba dengan merubah nilai dari thresholdnya, namun didapat hasil yang hampir sama.

Selanjutnya dilakukan proses penentuan acuan garis koordinat x dan y pada kolom dan baris. Karena hasil proses thresholding titik dot pada sisi 1 dan sisi 2 masih terlihat, maka berakibat pada proses pembacaan koordinat ini mengalami kesalahan juga. Hal tersebut dikarenakan titik dot hitam pada sisi 1 dan titik dot hitam bagian belakang dari sisi 2 sama sama terlihat. Sehingga dianggap oleh sistem sebagai titik dot hitam dalam satu sisi.

Selanjutnya dilakukan proses segmentasi untuk menentukan area setiap satu huruf braille. Karena pada proses penentuan garis koordinat x dan y mengalami kesalahan maka pada proses segmentasi ini juga mengalami kesalahan, yaitu area segmentasi yang dibentuk oleh sistem tidak tepat pada setiap huruf braille. Hasil proses segmentasi seperti terlihat pada Gambar 4.22, pada gambar tersebut terlihat area segmentasi yang tidak tepat pada area setiap huruf braille, dan hal tersebut merupakan kesalahan yang nantinya akan berakibat kesalahan dalam proses pengenalan.



Gambar 4.22 Hasil segmentasi pada proses pengenalan huruf braille pada cetakan 2 sisi

Dikarenakan dari proses segmentasi area tidak tepat pada setiap area satu huruf braille maka hasil pengenalan juga tidak benar. Seperti terlihat pada data pengujian di Tabel 4.11 hampir 100% hasil pengenalan tidak sesuai. Maka dapat disimpulkan bahwa sistem yang dibuat jika digunakan untuk proses pengenalan huruf braille pada cetakan 2 sisi hasilnya tidak akurat.

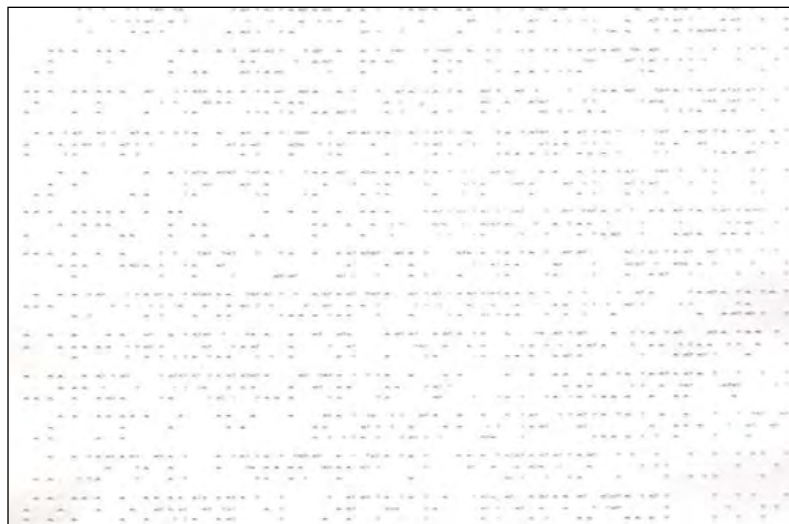
TABEL 4.11. Data pengujian pengenalan sistem untuk pengenalan huruf braille gambar dari hasil scanner pada cetakan 2 sisi

Percobaan	Huruf sebenarnya	Terbaca	Tidak terbaca	Kesalahan (%)	Akurasi (%)
1	443	2	441	99,55 %	0,45 %
2	443	2	441	99,55 %	0,45 %
3	443	1	442	99,78 %	0,22 %
4	443	2	441	99,55 %	0,45 %
5	443	2	441	99,55 %	0,45 %
6	443	2	441	99,55 %	0,45 %
7	443	1	442	99,78 %	0,22 %
8	443	3	440	99,33 %	0,67 %
9	443	2	441	99,55 %	0,45 %
10	443	2	441	99,55 %	0,45 %
Rata – rata tingkat akurasi					0,45 %

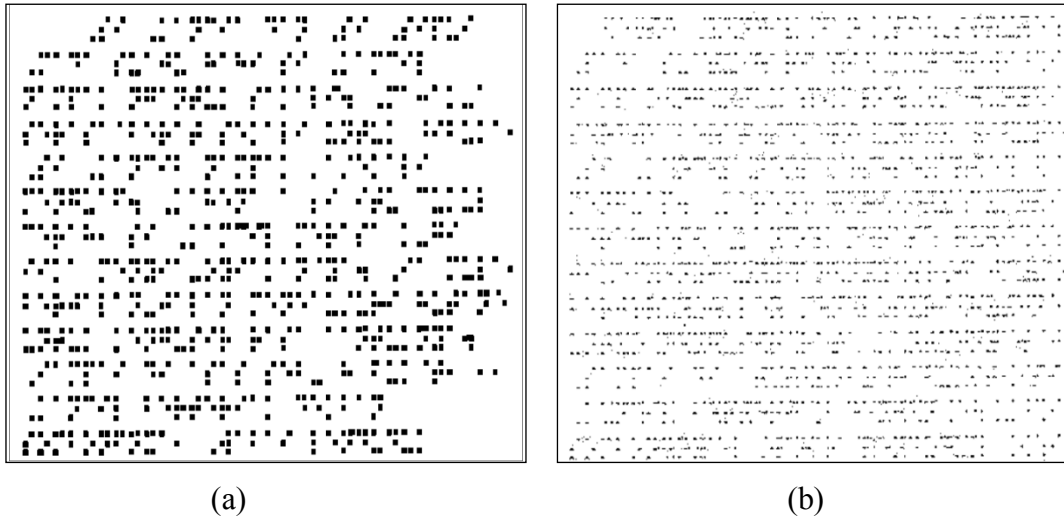
4.8 Pengujian Sistem Pengenalan Huruf Braille Pada Cetakan Double Side (2 Sisi) Pengambilan Gambar dengan Alat Scanner

Karena percobaan pengenalan huruf braille pada cetakan 2 sisi dengan pengambilan gambar menggunakan kamera webcam tidak berhasil dengan baik, maka dilakukan percobaan lagi dengan pengambilan gambar menggunakan alat scanner. Pengambilan gambar menggunakan alat scanner diharapkan akan dihasilkan gambar yang memiliki intensitas nilai piksel yang sama, sehingga akan memudahkan dalam pemisahan titik pada sisi 1 dan sisi 2 pada proses thresholding.

Pengambilan gambar dilakukan dengan cara men-scan lembar kertas huruf braille cetakan 2 sisi menggunakan alat scanner. Alat scanner yang digunakan merk Epson tipe WorkForce WF7511. Pada saat proses scanner digunakan pemilihan resolusi sebesar 75 dpi. Gambar hasil pengambilan gambar cetakan huruf braille pada cetakan 2 sisi seperti terlihat pada Gambar 4.23. Pada hasil gambar tersebut titik timbul pada sisi 1 dapat terlihat dan bagian belakang pada titik timbul sisi 2 juga terlihat, namun cenderung kelihatan lebih kecil. Diharapkan dari hasil pengambilan dengan alat scanner ini pada proses thresholding dapat memisahkan titik hitam sisi 1 dan titik hitam pada sisi 2.



Gambar 4.23 Hasil pengambilan gambar braille cetakan 2 sisi dengan alat scanner

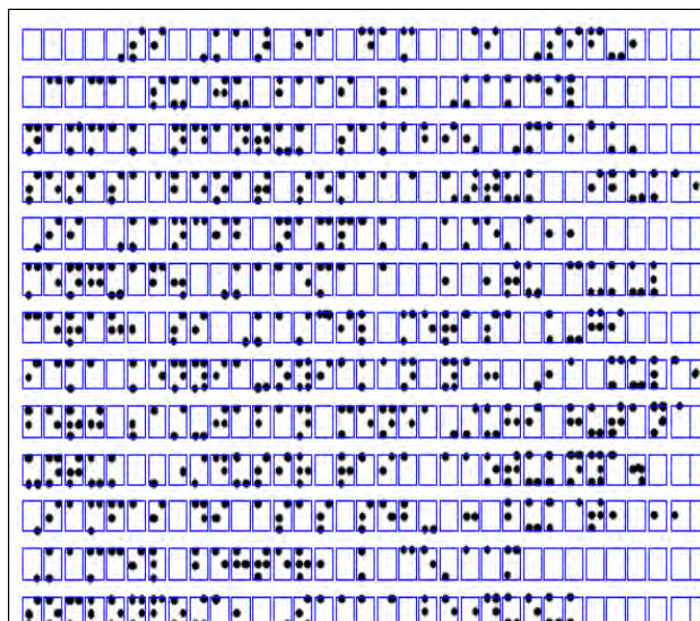


Gambar 4.24 (a). Hasil general threshold, (b) Hasil adaptive threshold

Pada proses thresholding dicoba dengan menggunakan General Threshold dan Adaptive Threshold. Proses dengan General Threshold didapatkan hasil yang lebih baik, titik belakang dari sisi ke 2 sudah tidak terlihat, namun ada beberapa titik hitam yang berada pada sisi 1 juga ikut tidak terlihat, seperti terlihat pada Gambar 4.24(a). Pada proses General Threshold digunakan nilai threshold 195, dicoba dengan menambah nilai threshold namun didapat hasil titik hitam pada sisi 2 jadi terlihat, jika nilai threshold di kurangi terlalu banyak, maka hasilnya banyak titik dot hitam yang hilang atau tidak terlihat.

Selanjutnya dilakukan proses penentuan acuan garis koordinat x dan y pada kolom dan baris. Karena hasil proses thresholding sudah bisa memisahkan titik dot pada sisi 1 dengan titik dot sisi 2, maka pada proses pembacaan koordinat ini mengalami didapatkan hasil yang lebih baik dibandingkan saat pengambilan gambar braille dengan kamera webcam.

Selanjutnya dilakukan proses segmentasi untuk menentukan area setiap satu huruf braille. Karena pada proses penentuan garis koordinat x dan y lebih baik, maka pada proses segmentasi ini juga menjadi lebih baik, yaitu area segmentasi yang dibentuk oleh sistem sudah tepat pada setiap huruf braille. Seperti terlihat pada Gambar 4.25.



Gambar 4.25. Hasil segmentasi pada proses pengenalan huruf braille pada cetakan 2 sisi gambar dari hasil scanner

Dikarenakan dari proses segmentasi area sudah tepat pada setiap area satu huruf braille maka hasil pengenalan juga sudah benar, namun karena pada proses thresholding beberapa titik dot hitam pada sisi 1 ada yang hilang maka hasil pengenalan juga masih tidak akurat.

TABEL 4.11. Data pengujian pengenalan sistem untuk pengenalan huruf braille gambar dari hasil scanner pada cetakan 2 sisi

Percobaan	Huruf sebenarnya	Terbaca	Tidak terbaca	Kesalahan (%)	Akurasi (%)
1	443	156	287	64,79 %	35,21 %
2	443	155	288	65,02 %	34,98 %
3	443	155	288	65,02 %	34,98 %
4	443	156	287	64,79 %	35,21 %
5	443	156	287	64,79 %	35,21 %
6	443	156	287	64,79 %	35,21 %
7	443	158	285	64,34 %	35,66 %
8	443	156	287	64,79 %	35,21 %
9	443	156	287	64,79 %	35,21 %
10	443	156	287	64,79 %	35,21 %
Rata – rata tingkat akurasi					35,20 %

Seperti terlihat pada Tabel 4.12 hasil data pengujian pengenalan huruf braille pada cetakan 2 sisi gambar dari scanner rata-rata tingkat akurasinya hanya 35,20%. Namun hasil tersebut lebih baik jika dibandingkan dengan pengambilan gambar dengan kamera webcam. Maka dapat disimpulkan bahwa sistem yang dibuat belum mampu untuk digunakan pengenalan huruf braille pada cetakan 2 sisi (double side).

BAB 5

PENUTUP

Dari hasil perancangan, pembuatan serta pengujian sistem pada penelitian dengan judul “PENGENALAN HURUF BRAILLE DENGAN MENGGUNAKAN METODE BLOB ANALYSIS DAN ARTIFICIAL NEURAL NETWORK” ini, penulis mendapatkan kesimpulan serta memberikan saran yang akan berguna bagi pengembangan penelitian ini.

5.1 Kesimpulan

Pada pengambilan gambar braille harus tersedia pencahayaan yang cukup terang dan konstan serta penempatan dari sumber cahaya diarahkan dari sisi samping dari kertas braille dengan sudut kemiringan 22° untuk mendapatkan gambar yang baik hasil capture dari webcam. Penggunaan teknik blob analysis sangat membantu dalam penentuan koordinat piksel dari setiap titik dot hitam, sehingga proses menentukan area segmentasi setiap huruf braille dapat dilakukan dengan tepat.

Pengenalan huruf braille dengan metode Artificial Neural Network dapat dihasilkan tingkat akurasi yang tinggi yaitu mencapai 99%. Sistem dapat melakukan proses pengenalan huruf braille pada gambar huruf braille dengan tingkat derajat kemiringan dari -1 derajat sampai 1 derajat. Tingkat akurasi sistem pengenalan mulai menurun saat derajat kemiringan gambar mulai 1,25 derajat, dan sistem tidak mampu mengenali sama sekali saat derajat kemiringan gambar pada 1.5 derajat.

Sistem belum mampu digunakan untuk proses pengenalan huruf braille pada cetakan 2 sisi (double side), walaupun pengambilan gambar dengan bantuan alat scanner terlebih dahulu.

5.2 Saran

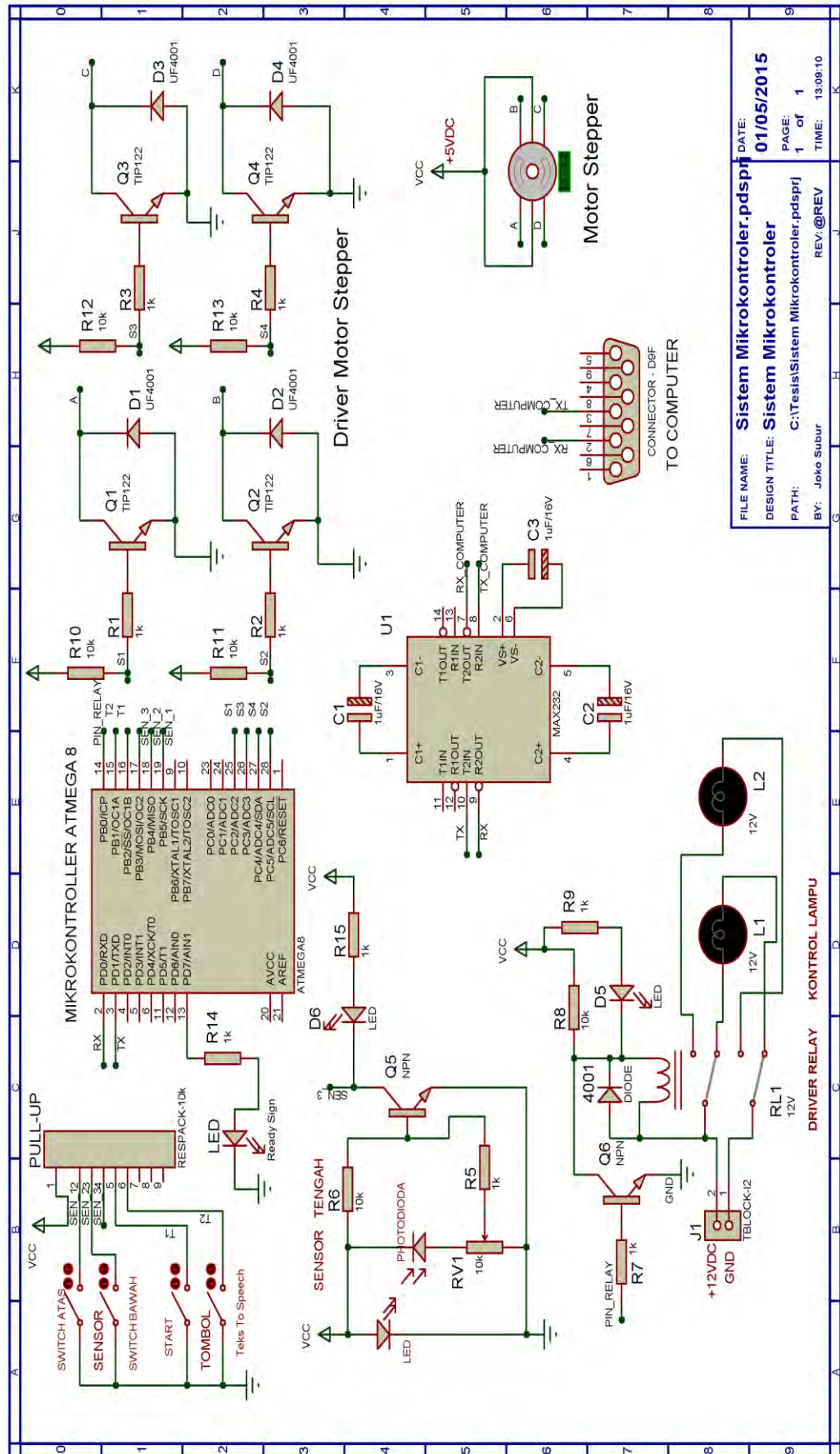
Untuk pengembangan selanjutnya bisa ditambahkan dengan output suara dengan dialek bahasa Indonesia. Dan dapat dikembangkan juga pada cetakan kertas braille double side (dua sisi) untuk mendapatkan hasil yang lebih baik.

(Halaman ini sengaja dikosongkan)

DAFTAR PUSTAKA

- Jie Li, Xiaoguang Yan,. (2010), "*Optical Braille Character Recognition with Support-Vector Machine Classifier*". International Conference on Computer Application and System Modeling (ICCASM 2010).
- Namba dan Zhang,. (2006), "*Cellular Neural Network for Associative Memory and Its Application to Braille Image Recognition*", International Joint Conference on Neural Networks, BC, Canada.
- Shreekanth, T dan Udayashankara, V,. (2013)," *A Review on Software Algorithms for Optical Recognition of Embossed Braille Characters*", International Journal of Computer Applications (0975-8887), volume 81-No.3.
- Sonka Milan, Hlavac Vaclav and Boyle Roger, "*Image Processing, Analisis dan Machine Vision*", Third Edition, Thomson Learning, USA, 2008.
- Wajid, M., Abdullah, M., Farooq, W.O.,. (2011), "*Imprinted Braille-Character Pattern Recognition using Image Processing Techniques*", International Conference on Image Information Processing.
- Wong, L., Abdulla, W., Hussmann, S. (2004), "*A Software Algorithm Prototype for Optical Recognition of Embossed Braille*", IEEE-2004. In: 17th Conference of the International Conference in Pattern Recognition, Cambridge, UK, pp. 23–26.
- Kusumadewi, S, Hartati, S,. (2006), "*Integrasi Sistem Fuzzy dan Jaringan Syaraf Tiruan*". ISBN: 979-756-124-1, Graha Ilmu, Yogyakarta.
- Munir, R,. (2004), "*Pengolahan Citra Digital*". ISBN: 979-3338-29-6, Cetakan pertama Agustus 2004, Informatika, Bandung.
- OpenCV Reference Manual, v2.2, Desember 2010

(Halaman ini sengaja dikosongkan)



(Halaman ini sengaja dikosongkan)

LAMPIRAN

LISTING PROGRAM

1. Program Penggerak Motor Stepper Untuk Mengatur Posisi Webcam dengan Bascom AVR

```
$regfile = "m8def.dat"      'Ic yg digunakan ATmega 8
$crystal = 11059200         'Nilai kristal 11,0592 MHz
$baud = 9600
=====
$hwstack = 32      'use 32 for the hardware stack
$swstack = 10      'default use 10 for the SW stack
$framesize = 40    'default use 40 for the frame space
=====
Config Portc.2 = Output  'Inisialisasi Port
Config Portc.3 = Output  'Inisialisasi Port
Config Portc.4 = Output  'Inisialisasi Port
Config Portc.5 = Output  'Inisialisasi Port
=====
B Alias Portc.5
D Alias Portc.4
C Alias Portc.3
A Alias Portc.2
=====
Config Portb.0 = Output
Pin_relay Alias Portb.0
=====
Config Pinb.1 = Input
Config Pinb.2 = Input
Config Pinb.3 = Input
Config Pinb.4 = Input
Config Pinb.5 = Input
Config Portd.7 = Output
=====
Sensor_atas Alias Pinb.5
Sensor_bawah Alias Pinb.4
Sensor_tengah Alias Pinb.3
=====
Tombol_start Alias Pinb.2
Tombol_voice Alias Pinb.1
=====
Led_siap Alias Portd.7
=====
Dim Ulang As Byte
Dim Terima As Byte
Dim Hitung As Byte
Dim Status_siap As Byte
=====
'Konfigurasi Pin Antara Mikro dengan Motor Stepper
' 7 6 5 4 3 2 1 0 >> Bit Mikro
'- - B D C A - - >> Kabel Motor Stepper
Const Tunda = 5      'dalam mili detik
Portc = &H00
Waitms 100          'tunda 100 milidetik
Awal:
Pin_relay = 1      'aktifkan relay (lampu bawah nyala)
Led_siap = 1      'led siap aktif high (0=mati, 1=nyala)
Sensor_atas = 1
Sensor_bawah = 1
Sensor_tengah = 1
Tombol_start = 1
Tombol_voice = 1
Status_siap = 1
Waitms 100
Gosub Posisi_awal
Waitms 100
Do
  Terima = Inkey()
  If Terima <> 0 Then
    If Terima = "A" Then
      Pin_relay = 1      'aktifkan relay (lampu bawah nyala)
      Gosub Posisi_atas
```

```

Elseif Terima = "B" Then
    Pin_relay = 0 'aktifkan relay (lampu atas nyala)
    Gosub Posisi_tengah
Elseif Terima = "C" Then
    Pin_relay = 0 'aktifkan relay (lampu atas nyala)
    Gosub Posisi_bawah
Elseif Terima = "D" Then
    Status_siap = 1
    Led_siap = 1 'led siap nyala
Elseif Terima = "E" Then
    Status_siap = 0
    Led_siap = 0 'led siap mati
Elseif Terima = "F" Then
    Gosub Posisi_kembali_awal
    Led_siap = 0 'led siap mati
End If
End If
'=====
If Status_siap = 1 Then
    If Tombol_start = 0 Then
        Print "s";
        Wait 1
        Print "##";
        Waitms 100
        Bitwait Tombol_start , Set
    Elseif Tombol_voice = 0 Then
        Print "v";
        Wait 1
        Print "##";
        Waitms 100
        Bitwait Tombol_voice , Set
    End If
End If
Loop
'=====
Posisi_awal:
Sensor_atas = 1
'===== PUTAR KIRI =====
While Sensor_atas = 0
    A = 1 : B = 0 : C = 0 : D = 0
    Waitms Tunda
    A = 1 : B = 1 : C = 0 : D = 0
    Waitms Tunda
'=====
    A = 0 : B = 1 : C = 0 : D = 0
    Waitms Tunda
    A = 0 : B = 1 : C = 1 : D = 0
    Waitms Tunda
'=====
    A = 0 : B = 0 : C = 1 : D = 0
    Waitms Tunda
    A = 0 : B = 0 : C = 1 : D = 1
    Waitms Tunda
'=====
    A = 0 : B = 0 : C = 0 : D = 1
    Waitms Tunda
    A = 1 : B = 0 : C = 0 : D = 1
    Waitms Tunda
'=====
Wend
Return
'=====

Posisi_kembali_awal:
Sensor_atas = 1
'===== PUTAR KIRI =====
While Sensor_atas = 0
    A = 1 : B = 0 : C = 0 : D = 0
    Waitms Tunda
    A = 1 : B = 1 : C = 0 : D = 0
    Waitms Tunda
'=====
    A = 0 : B = 1 : C = 0 : D = 0

```

```

Waitms Tunda
A = 0 : B = 1 : C = 1 : D = 0
Waitms Tunda
'====

A = 0 : B = 0 : C = 1 : D = 0
Waitms Tunda
A = 0 : B = 0 : C = 1 : D = 1
Waitms Tunda
'====

A = 0 : B = 0 : C = 0 : D = 1
Waitms Tunda
A = 1 : B = 0 : C = 0 : D = 1
Waitms Tunda
'====

Wend
Print "d";
Wait 1
Pin_relay = 1 'aktifkan relay (lampu bawah nyala)
Print "#";
Waitms 100
Return
'=====

Posisi_bawah:
'===== PUTAR KANAN sambil baca sensor switch =====
While Sensor_bawah = 0
    A = 0 : B = 0 : C = 0 : D = 1
    Waitms Tunda
    A = 0 : B = 0 : C = 1 : D = 1
    Waitms Tunda
    '====

    A = 0 : B = 0 : C = 1 : D = 0
    Waitms Tunda
    A = 0 : B = 1 : C = 1 : D = 0
    Waitms Tunda
    '====

    A = 0 : B = 1 : C = 0 : D = 0
    Waitms Tunda
    A = 1 : B = 1 : C = 0 : D = 0
    Waitms Tunda
    '====

    A = 1 : B = 0 : C = 0 : D = 0
    Waitms Tunda
    A = 1 : B = 0 : C = 0 : D = 1
    Waitms Tunda
    '====

Wend
Wait 1
Print "c";
Wait 1
Print "#";
Waitms 100
Return
'=====

Posisi_tengah:
'===== Saat Posisi Acak =====
If Sensor_atas = 0 And Sensor_bawah = 0 Then
    '===== Putar Kiri Menuju Ke Posisi Atas Dahulu =====
    While Sensor_atas = 0
        A = 1 : B = 0 : C = 0 : D = 0
        Waitms Tunda
        A = 1 : B = 1 : C = 0 : D = 0
        Waitms Tunda
        '====

        A = 0 : B = 1 : C = 0 : D = 0
        Waitms Tunda
        A = 0 : B = 1 : C = 1 : D = 0
        Waitms Tunda
        '====

        A = 0 : B = 0 : C = 1 : D = 0
        Waitms Tunda
        A = 0 : B = 0 : C = 1 : D = 1
        Waitms Tunda
        '====

```

```

A = 0 : B = 0 : C = 0 : D = 1
Waitms Tunda
A = 1 : B = 0 : C = 0 : D = 1
Waitms Tunda
'====
Wend
Waitms 200
'===== Putar Kanan Menuju Posisi Tengah =====
While Sensor_tengah = 1
  A = 0 : B = 0 : C = 0 : D = 1
  Waitms Tunda
  A = 0 : B = 0 : C = 1 : D = 1
  Waitms Tunda
  '====
  A = 0 : B = 0 : C = 1 : D = 0
  Waitms Tunda
  A = 0 : B = 1 : C = 1 : D = 0
  Waitms Tunda
  '====
  A = 0 : B = 1 : C = 0 : D = 0
  Waitms Tunda
  A = 1 : B = 1 : C = 0 : D = 0
  Waitms Tunda
  '====
  A = 1 : B = 0 : C = 0 : D = 0
  Waitms Tunda
  A = 1 : B = 0 : C = 0 : D = 1
  Waitms Tunda
  '====
Wend
Wait 1
Print "b";
Wait 1
Print "#";
Waitms 100
'Posisi sudah mepet sensor bawah
Elseif Sensor_atas = 0 And Sensor_bawah = 1 Then
'===== Putar Kiri Menuju Tengah =====
While Sensor_tengah = 1
  A = 1 : B = 0 : C = 0 : D = 0
  Waitms Tunda
  A = 1 : B = 1 : C = 0 : D = 0
  Waitms Tunda
  '====
  A = 0 : B = 1 : C = 0 : D = 0
  Waitms Tunda
  A = 0 : B = 1 : C = 1 : D = 0
  Waitms Tunda
  '====
  A = 0 : B = 0 : C = 1 : D = 0
  Waitms Tunda
  A = 0 : B = 0 : C = 1 : D = 1
  Waitms Tunda
  '====
  A = 0 : B = 0 : C = 0 : D = 1
  Waitms Tunda
  A = 1 : B = 0 : C = 0 : D = 1
  Waitms Tunda
  '====
Wend
Wait 1
Print "b";
Wait 1
Print "#";
Waitms 100
'Posisi sudah mepet sensor atas
Elseif Sensor_atas = 1 And Sensor_bawah = 0 Then
'===== Putar Kanan Menuju Tengah =====
While Sensor_tengah = 1
  A = 0 : B = 0 : C = 0 : D = 1
  Waitms Tunda
  A = 0 : B = 0 : C = 1 : D = 1
  Waitms Tunda

```

```

=====
A = 0 : B = 0 : C = 1 : D = 0
Waitms Tunda
A = 0 : B = 1 : C = 1 : D = 0
Waitms Tunda
=====
A = 0 : B = 1 : C = 0 : D = 0
Waitms Tunda
A = 1 : B = 1 : C = 0 : D = 0
Waitms Tunda
=====
A = 1 : B = 0 : C = 0 : D = 0
Waitms Tunda
A = 1 : B = 0 : C = 0 : D = 1
Waitms Tunda
=====
Wend
Wait 1
Print "b";
Wait 1
Print "#";
Waitms 100
' Posisi sudah berada di tengah
Elseif Sensor_tengah = 0 Then
    Wait 1
    Print "b";
    Wait 1
    Print "#";
    Waitms 100
End If
Return
=====
Posisi_atas:
===== Putar Kiri sambil baca sensor switch =====
While Sensor_atas = 0
    A = 1 : B = 0 : C = 0 : D = 0
    Waitms Tunda
    A = 1 : B = 1 : C = 0 : D = 0
    Waitms Tunda
    =====
    A = 0 : B = 1 : C = 0 : D = 0
    Waitms Tunda
    A = 0 : B = 1 : C = 1 : D = 0
    Waitms Tunda
    =====
    A = 0 : B = 0 : C = 1 : D = 0
    Waitms Tunda
    A = 0 : B = 0 : C = 1 : D = 1
    Waitms Tunda
    =====
    A = 0 : B = 0 : C = 0 : D = 1
    Waitms Tunda
    A = 1 : B = 0 : C = 0 : D = 1
    Waitms Tunda
    =====
Wend
Wait 1
Print "a";
Wait 1
Print "#";
Waitms 100
Return
End          'end program

```

2. Program Antar Muka Pada Komputer-Visual Studio 2010-OpenCV

```
#pragma once
#include <cv.h>
#include <cxcvcore.h>
#include <highgui.h>
#include <time.h>
#include <iostream>
#include <cmath>
#include <string.h>
#include <stdlib.h>
#include <windows.h>
#include <mmsystem.h>
using <microsoft.dll>

#define PI 3.14159265
namespace OpenCV_Basic
{
    using namespace std;
    using namespace System::IO::Ports;
    using namespace System;
    using namespace System::ComponentModel;
    using namespace System::Collections;
    using namespace System::Windows::Forms;
    using namespace System::Data;
    using namespace System::Drawing;
    using namespace System::Runtime::InteropServices;
    using namespace System::Speech::Synthesis;
    using namespace System::Runtime::InteropServices; // Untuk convert string to
char
    using namespace System::IO; //for file or text management

IplImage* origimage=0;
IplImage* origimage1=0;
IplImage* origimage2=0;
IplImage* origimage3=0;
IplImage* origimage4=0;
IplImage* origimage5=0;
IplImage* origimage6=0;
IplImage* hasil_crop=0;
IplImage* hasil_crop1=0;
IplImage* hasil_crop2=0;
IplImage* hasil_crop3=0;
IplImage* hasil_crop4=0;
IplImage* hasil_crop5=0;
IplImage* hasil_crop6=0;
IplImage* hasil_putar=0;
IplImage* hasil_putar1=0;
IplImage* hasil_putar2=0;
IplImage* hasil_putar3=0;
IplImage* hasil_putar4=0;
IplImage* hasil_putar5=0;
IplImage* hasil_putar6=0;
IplImage* erot=0;
IplImage* titik_contour=0;
IplImage* dilation=0;
IplImage* tampung_erot=0;
IplImage* image_miring=0;
IplImage* grayimage;
IplImage* grayimage1;
IplImage* grayimage2;
IplImage* grayimage3;
IplImage* grayimage4;
IplImage* grayimage5;
IplImage* grayimage6;
IplImage* binaryimage;
IplImage* binaryimage1;
IplImage* binaryimage2;
IplImage* binaryimage3;
IplImage* binaryimage4;
```

```

IplImage *binaryimage5;
IplImage *binaryimage6;
IplImage* imagecontour_1=0;
IplImage* imagecontour_2=0;
IplImage* imagecontour_3=0;
IplImage* imagecontour_4=0;
IplImage* imagecontour_5=0;
IplImage* imagecontour_6=0;
IplImage* imagecontour_Scan=0;
IplImage *titik_contour_1=0;
IplImage *titik_contour_2=0;
IplImage *titik_contour_3=0;
IplImage *titik_contour_4=0;
IplImage *titik_contour_5=0;
IplImage *titik_contour_6=0;
IplImage *titik_contour_Scan=0;
IplImage *segmentasi_1=0;
IplImage *segmentasi_2=0;
IplImage *segmentasi_3=0;
IplImage *segmentasi_4=0;
IplImage *segmentasi_5=0;
IplImage *segmentasi_6=0;
IplImage* tampung_1=0;
IplImage* tampung_2=0;
IplImage* tampung_3=0;
IplImage* tampung_4=0;
IplImage* tampung_5=0;
IplImage* tampung_6=0;
IplImage* tampil_1=0;
IplImage* tampil_2=0;
IplImage* tampil_3=0;
IplImage* tampil_4=0;
IplImage* tampil_5=0;
IplImage* tampil_6=0;
IplImage* tampil_Scan=0;
IplImage* biner2rgb=0;
IplImage* gray2rgb=0;
IplImage* gray2rgb_1=0;
IplImage* threshold2rgb=0;
IplImage* erot2rgb=0;
IplImage* dilation2rgb=0;
IplImage* segmentasi2rgb=0;
IplImage* image_full=0;

//utk kamera
CvCapture* capture = 0;
CvCapture* capture1 = 0;
CvCapture* capture2 = 0;
IplImage *frame=0;
IplImage *frame1=0;
IplImage *frame2=0;
IplImage *tampungframe=0;
IplImage *tampungframe1=0;
IplImage *tampungframe2=0;
IplImage *bantutampung=0;
IplImage* processimage=0;
int start_stop=0;
int status_sumber=0;    // pilih sumber dari file gambar atau Kamera, 0=file gambar
                          1=kamera
int pilih_threshold=1;  // pilih jenis threshold. 0=threshold biasa(general), 1=adaptive
                          threshold.
int pilih_operasi=1;    //pilih jenis operasi. 0>manual, 1=otomatis.
int kapital=0,tanda_angka=0;
int simpanx=10000,simpany=10000;
int simpanx_bantu=10000;
int simpany_bantu=10000;
int xx,yy,i,j,a,j2;
int banyak_erot=1;
int hitam=0,putih=0;
int lebar_min=0,lebar_max=0
int hitung_lebar=0;

```



```

int tinggi_min=0,tinggi_max=0;
int tinggi=0;
int kode1=0,kode2=0,kode3=0;
int kode4=0,kode5=0,kode6=0;
int hitung_capture=0;
int persiapan=0;

// untuk perhitungan kemiringan
double A=0,B=0,C=0;
double aa=0,bb=0,cc=0;
double sin_A=0,sin_B=0,sin_C=0;
CvScalar* warna = 0;
char huruf_hasil;
CvPoint sudut1,sudut2;
CvPoint garis1,garis2;
double derajat1=1.7;
double derajat2=-1;

//== Data pixel proses CROP
int Yawal1=45,Yakhir1=400,Xawal1=50,Xakhir1=505;
int Yawal2=45,Yakhir2=400,Xawal2=50,Xakhir2=505;
int Yawal3=45,Yakhir3=400,Xawal3=50,Xakhir3=505;
int awal4=45,Yakhir4=400,Xawal4=50,Xakhir4=505;
int Yawal5=45,Yakhir5=400,Xawal5=50,Xakhir5=505;
int Yawal6=45,Yakhir6=400,Xawal6=50,Xakhir6=505;

// UNTUK PROSES BLOB ANALYSIS
int count=0,tt=0;
CvPoint titik_pusat;
CvPoint titik_pusat1;
CvPoint titik_pusat2;
CvPoint titik_pusat3;
CvPoint titik_pusat4;
CvPoint titik_pusat5;
CvPoint titik_pusat6;
CvPoint titik_pusatScan;
//===
CvPoint tanda_merah;
CvPoint tanda_merah1;
CvPoint tanda_merah2;
CvPoint tanda_merah3;
CvPoint tanda_merah4;
CvPoint tanda_merah5;
CvPoint tanda_merah6;
CvPoint tanda_merahScan;
//===
// image 1
int count1=0;
int rata_x1[1000],rata_y1[1000];
int titik_x1[1000],titik_y1[1000];
int garis_x1[1000],garis_y1[1000];
// image 2
int count2=0;
int rata_x2[1000],rata_y2[1000];
int titik_x2[1000],titik_y2[1000];
int garis_x2[1000],garis_y2[1000];
// image 3
int count3=0;
int rata_x3[1000],rata_y3[1000];
int titik_x3[1000],titik_y3[1000];
int garis_x3[1000],garis_y3[1000];
// image 4
int count4=0;
int rata_x4[1000],rata_y4[1000];
int titik_x4[1000],titik_y4[1000];
int garis_x4[1000],garis_y4[1000];
// image 5
int count5=0;
int rata_x5[1000],rata_y5[1000];
int titik_x5[1000],titik_y5[1000];
int garis_x5[1000],garis_y5[1000];

```

```

// image 6
int count6=0;
int rata_x6[1000],rata_y6[1000];
int titik_x6[1000],titik_y6[1000];
int garis_x6[1000],garis_y6[1000];
//====
// image Scan
int countScan=0;
int rata_xScan[1000],rata_yScan[1000];
int titik_xScan[1000],titik_yScan[1000];
int garis_xScan[1000],garis_yScan[1000];
//====
int startx,starty;
int startx1,starty1;
int startx2,starty2;
int startx3,starty3;
int startx4,starty4;
int startx5,starty5;
int startx6,starty6;
int startxScan,startyScan;
//untuk segmentasi
int kolom=0,baris=0;
int kolom1=0,baris1=0;
int kolom2=0,baris2=0;
int kolom3=0,baris3=0;
int kolom4=0,baris4=0;
int kolom5=0,baris5=0;
int kolom6=0,baris6=0;
int kolomScan=0,barisScan=0;
//====
int koordinat_x[1000],koordinat_y[1000];
int koordinat_x1[1000],koordinat_y1[1000];
int koordinat_x2[1000],koordinat_y2[1000];
int koordinat_x3[1000],koordinat_y3[1000];
int koordinat_x4[1000],koordinat_y4[1000];
int koordinat_x5[1000],koordinat_y5[1000];
int koordinat_x6[1000],koordinat_y6[1000];
int koordinat_xScan[1000],koordinat_yScan[1000];

// Untuk Jaringan Syaraf Tiruan
double WL1 [100][100];
double WL2 [100][100];
double WL3 [100][50];
double WLOUT [50][50];
double BIAS [10][100];
double WL1_UPDATE [100][100];
double WL2_UPDATE [100][100];
double WL3_UPDATE [100][50];
double WLOUT_LAMA [50][50];
double BIAS_UPDATE [10][100];
//====
double HASIL_LAYER [10][100];
double HASIL_G [10][100];
double DELTA [10][100];
double DELTA_UPDATE [10][100];
double Y [10][100];
double Y0 [10][100];
double Y_OUT [10];
double OUT_ERROR [100][100];
double SSE [100];
double tamp [100];
double alfa1 [100];
double alfa2 [100];
double alfa3 [500];
int XIN [200][100];
int XIN0 [200][100];
int X_TRAIN [200];
int TARGET [200][100];
int BACA_DATA [200][100];
//=====
long iterasi=0;

```

```

double   miu_nol=0;
double   k_nol=0;
double   learn_rate=0;
double   tao=0;
double   miu=0;
double   alpha=0;
double   alf=0.9;
double   beta=0;
double   ENERGI=0;
double   ERROR_AKHIR=0;
double   error_limit=0;
double   MSE=0;
double   temp=0;
int       node_input=0;
int       node_hidden1=0;
int       node_hidden2=0;
int       node_output=0;
int       banyak_data=0;
int       banyak_input=0;
int       polake=0;
int       speedX=1;
double   thresx = 0.75;
bool      status_learning=0;
//=====
char huruf;
int jenis_cetakan=1;    //1=1 sisi ; 2=2sisi
/// <summary>
/// Summary for Form1

#pragma endregion
private: System::Void button_exit_Click_1(System::Object^ sender, System::EventArgs^ e)
{
    Close(); }

private: System::Void comboBox1_SelectedIndexChanged(System::Object^ sender,
System::EventArgs^ e) {
    if (comboBox1->Text=="Kamera") {
        // Mengaktifkan kamera
        //====
        //tampilkan video dari webcam 1
        capture1 = cvCaptureFromCAM(1);
        frame1 = cvQueryFrame( capture1 );
        //====
        if(tampungframe1 == 0)
        {
            tampungframe1=cvCreateImage( cvSize(frame1->width,frame1->height), 8, 1 );
        }
        if(tampungframe1 != 0)
        {
            cvReleaseImage( &tampungframe1 );
            tampungframe1=cvCreateImage( cvSize(frame1->width,frame1->height), 8, 1 );
        }
        tampungframe1=cvCloneImage(frame1);
        //tampilkan video dari webcam 2
        capture2 = cvCaptureFromCAM(2);
        frame2 = cvQueryFrame( capture2 );
        //====
        if(tampungframe2 == 0)
        {
            tampungframe2=cvCreateImage( cvSize(frame2->width,frame2->height), 8, 1 );
        }
        if(tampungframe2 != 0)
        {
            cvReleaseImage( &tampungframe2 );
            tampungframe2=cvCreateImage( cvSize(frame2->width,frame2->height), 8, 1 );
        }
        //====
        tampungframe2=cvCloneImage(frame2);
        tabControl1->SelectTab("tabPage1");
        //====
    }
}

```

```

button2->Enabled=0;
groupBox7->Enabled=0;
groupBox6->Enabled=1;
button_capture->Enabled=1;
status_sumber=1;
pilih_operasi=1;
serialPort1->Open();
serialPort1->WriteLine("D"); // Status siap pada mikro diubah ke 1
comboBox1->Text="Kamera";
label_keterangan->Text="Ready";
button4->Enabled=true; // button Posisi Atas
button5->Enabled=true; // button Posisi Tengah
button6->Enabled=true; // button Posisi Bawah
button8->Enabled=true; // button close port
timer_cek->Start();
timer_camera->Start();
timer_camera2->Start();
}
// memilih file Gambar
if (comboBox1->Text=="File Gambar")
{
    // Menghentikan kamera
    tabControl1->SelectTab("tabPage1");
    groupBox7->Enabled=1;
    groupBox6->Enabled=0;
    button2->Enabled=1;
    button_capture->Enabled=0;
    status_sumber=0;
    timer_camera->Stop();
    timer_camera2->Stop();
}
}

private: System::Void timer_camera_Tick
(System::Object^ sender, System::EventArgs^ e)
{
    if (status_sumber==1)
    {
        frame1 = cvQueryFrame( capture1 );
        cvCopy(frame1,tampungframe1,0);
        pictureBox_image1->Image = gcnew //pengganti cvShowImage untuk menampilkan gambar
        System::Drawing::Bitmap(tampungframe1->width,tampungframe1->height,tampungframe1-
        >widthStep,
        System::Drawing::Imaging::PixelFormat::Format24bppRgb,(System::IntPtr)tampungframe1-
        >imageData);
        pictureBox_image1->Refresh();
    }
};

// button capture gambar braille
private: System::Void button_capture_Click(System::Object^ sender, System::EventArgs^ e)
{
    timer_camera->Enabled=false;
    timer_camera2->Enabled=false;
    //===
    if (hitung_capture==1)
    {
        origimage1 = cvCreateImage(cvSize(tampungframe1->width, tampungframe1->height),
        tampungframe1->depth, 1);
        origimage1=cvCloneImage(tampungframe1);
        //===
        pictureBox_capture1->Image = gcnew //pengganti cvShowImage untuk menampilkan gambar
        System::Drawing::Bitmap(origimage1->width,origimage1->height,origimage1->widthStep,
        System::Drawing::Imaging::PixelFormat::Format24bppRgb,(System::IntPtr) origimage1-
        >imageData);
        pictureBox_capture1->Refresh();
        //=====
        origimage2 = cvCreateImage(cvSize(tampungframe2->width, tampungframe2->height),
        tampungframe2->depth, 1);
        origimage2=cvCloneImage(tampungframe2);
        //===

```

```

pictureBox_capture2->Image = gcnew System::Drawing::Bitmap(origimage2-
>width,origimage2->height,origimage2->widthStep,
System::Drawing::Imaging::PixelFormat::Format24bppRgb,(System::IntPtr) origimage2-
>imageData);
pictureBox_capture2->Refresh();
};
//===
if (hitung_capture==2)
{
origimage3 = cvCreateImage(cvSize(tampungframe1->width, tampungframe1->height),
tampungframe1->depth, 1);
origimage3=cvCloneImage(tampungframe1);
//===
pictureBox_capture3->Image = gcnew System::Drawing::Bitmap(origimage3-
>width,origimage3->height,origimage3-
>widthStep,System::Drawing::Imaging::PixelFormat::Format24bppRgb,(System::IntPtr)
origimage3->imageData); pictureBox_capture3->Refresh();
//=====
origimage4 = cvCreateImage(cvSize(tampungframe2->width, tampungframe2->height),
tampungframe2->depth, 1);
origimage4=cvCloneImage(tampungframe2);
//===
pictureBox_capture4->Image = gcnew System::Drawing::Bitmap(origimage4-
>width,origimage4->height,origimage4-
>widthStep,System::Drawing::Imaging::PixelFormat::Format24bppRgb,(System::IntPtr)
origimage4->imageData);
pictureBox_capture4->Refresh();
//=====
};
//===
if (hitung_capture==3)
{
origimage5 = cvCreateImage(cvSize(tampungframe1->width, tampungframe1->height),
tampungframe1->depth, 1);
origimage5=cvCloneImage(tampungframe1);
//===
pictureBox_capture5->Image = gcnew //pengganti cvShowImage untuk menampilkan gambar
System::Drawing::Bitmap(origimage5->width,origimage5->height,origimage5->widthStep,
System::Drawing::Imaging::PixelFormat::Format24bppRgb,(System::IntPtr) origimage5-
>imageData);
pictureBox_capture5->Refresh();
//=====
origimage6 = cvCreateImage(cvSize(tampungframe2->width, tampungframe2->height),
tampungframe2->depth, 1);
origimage6=cvCloneImage(tampungframe2);
//===
pictureBox_capture6->Image = gcnew //pengganti cvShowImage untuk menampilkan gambar
System::Drawing::Bitmap(origimage6->width,origimage6->height,origimage6->widthStep,
System::Drawing::Imaging::PixelFormat::Format24bppRgb,(System::IntPtr) origimage6-
>imageData);
pictureBox_capture2->Refresh();
//=====
};
timer_camera->Enabled=true;
timer_camera2->Enabled=true;
}

private: System::Void button_TTS_Click(System::Object^ sender, System::EventArgs^ e) {
    try
    {
        SpeechSynthesizer^ synth=gcnew SpeechSynthesizer();
        synth->Speak(richTextBox_hasil->Text);
    }
    catch (Exception^ ex)
    {
        SpeechSynthesizer^ synth=gcnew SpeechSynthesizer();
        synth->Speak("Error");
    }
    Sleep(1500);
    label_keterangan->Text="Ready";
    serialPort1->WriteLine("D"); // Status siap pada mikro diubah ke 1
}

```

```

    }

private: System::Void button_saveimage_Click(System::Object^ sender, System::EventArgs^ e) {
    System::String^ namafile = textBox_namafile->Text + ".jpg";
    IntPtr pointer_to_text_string = Marshal::StringToHGlobalAnsi(namafile);
    const char* string_for_opencv = static_cast<const char*>(pointer_to_text_string.ToPointer());
    int itung = int::Parse(textBox_C_Y1->Text);
    if (itung==1){cvSaveImage(string_for_opencv,origimage1);};
    if (itung==2){cvSaveImage(string_for_opencv,origimage2);};
    if (itung==3){cvSaveImage(string_for_opencv,origimage3);};
    if (itung==4){cvSaveImage(string_for_opencv,origimage4);};
    if (itung==5){cvSaveImage(string_for_opencv,origimage5);};
    if (itung==6){cvSaveImage(string_for_opencv,origimage6);};
}

private: System::Void timer_camera2_Tick(System::Object^ sender, System::EventArgs^ e) {
    if (status_sumber==1)
    {
        frame2 = cvQueryFrame( capture2 );
        cvCopy(frame2,tampungframe2,0);
        //====
        pictureBox_image2->Image = gcnew //pengganti cvShowImage untuk menampilkan gambar
        System::Drawing::Bitmap(tampungframe2->width,tampungframe2->height,tampungframe2->widthStep,
        System::Drawing::Imaging::PixelFormat::Format24bppRgb,(System::IntPtr)tampungframe2->imageData);
        pictureBox_image2->Refresh();
    };
}

private: System::Void Form1_Load(System::Object^ sender, System::EventArgs^ e) {
    hitung_capture=0;
    persiapan=0;
    CvRNG cvRNG(int64 seed=-1);
}

// Button Open POrt
private: System::Void button7_Click(System::Object^ sender, System::EventArgs^ e) {
    serialPort1->Open();
    button4->Enabled=true;
    button5->Enabled=true;
    button6->Enabled=true;
    button8->Enabled=true;
    button7->Enabled=false;
}

// Button Close POrt
private: System::Void button8_Click(System::Object^ sender, System::EventArgs^ e) {
    serialPort1->Close();
    button4->Enabled=false;
    button5->Enabled=false;
    button6->Enabled=false;
    button8->Enabled=false;
    button7->Enabled=true;
}

// Button kirim tanda siap
private: System::Void button3_Click(System::Object^ sender, System::EventArgs^ e) {
    serialPort1->WriteLine("S");
}

// Button kirim tanda geser ke atas
private: System::Void button4_Click(System::Object^ sender, System::EventArgs^ e) {
    serialPort1->WriteLine("A");
    label_ket_posisi->Text="Posisi: ATAS";
    hitung_capture=1;
}

```

```

}

// Button kirim tanda geser ke tengah
private: System::Void button5_Click(System::Object^ sender, System::EventArgs^ e) {
    serialPort1->WriteLine("B");
    label_ket_posisi->Text="Posisi: TENGAH";
    hitung_capture=2;
}

// Button kirim tanda geser ke Bawah
private: System::Void button6_Click(System::Object^ sender, System::EventArgs^ e) {
    serialPort1->WriteLine("C");
    label_ket_posisi->Text="Posisi: BAWAH";
    hitung_capture=3;
}

// Button TIMER CEK
private: System::Void timer_cek_Tick(System::Object^ sender, System::EventArgs^ e) {
    if (pilih_operasi==1) // jika pilih menu otomatis
    {
        if (terima_lama!=terima)
        {
            terima_lama=terima;
            if (terima=="s")
            {
                if (persiapan==0)
                {
                    persiapan=1;
                    tabControl1->SelectTab("tabPage1");
                }
                else
                {
                    persiapan=0;
                    Sleep(1500);
                    // Status siap pada mikro diubah ke 0
                    serialPort1->WriteLine("E");
                    hitung_capture=0;
                    // memulai proses
                    serialPort1->WriteLine("A");
                    label_ket_posisi->Text="Posisi: ATAS";
                    hitung_capture=1;
                    label_keterangan->Text="Processing. . .";
                    //=====
                }
            }
        }
        if (terima=="a")
        {
            hitung_capture=1;
            this->button_capture_Click(sender, e);
            this->button5_Click(sender, e); // klik tombol posisi tengah
        }
        if (terima=="b")
        {
            hitung_capture=2;
            this->button_capture_Click(sender, e);
            this->button6_Click(sender, e); // klik tombol posisi bawah
        }
        if (terima=="c")
        {
            hitung_capture=3;
            this->button_capture_Click(sender, e);
            tabControl1->SelectTab("tabPage6");
            timer_bantu->Enabled=true;
        }
        if (terima=="d")
        {
            this->btn_luruskan_auto_Click(sender, e);
        }
    }
}

```

```

if (terima=="v")           // v=Voice proses Teks to Speech
{
    label_keterangan->Text="Teks to Speech Processing";
    Sleep(500);
    serialPort1->WriteLine("E");    // Status siap pada mikro diubah ke 0
    this->button_TTS_Click(sender, e);
}
//=====
}
}

//===== MULAI PROSES ARTIFICIAL NEURAL NETWORK =====

// Button Set Bobot dan Bias
private: System::Void button17_Click(System::Object^ sender, System::EventArgs^ e) {
double nilai_random,bet;
int m=0,n=0,p=0;
//=====
banyak_data = int::Parse(textBox1->Text);
banyak_input = int::Parse(textBox2->Text);
node_hidden1 = int::Parse(textBox3->Text);
node_hidden2 = int::Parse(textBox37->Text);
node_output = int::Parse(textBox4->Text);
//=====
if (radio_bias_acak->Checked)
{
    // mengisi nilai semua bobot dan bias antara -0.5 sampai 0.5
    // Bobot input ke layer hidden1
    for (m = 1; m <= node_hidden1; m++)
    {
        for (n = 1; n <= banyak_input; n++)
        {
            nilai_random = rand()%1000;
            nilai_random = ((nilai_random / 1000) - 0.5);
            WL1 [m][n] = nilai_random;
        }
    }
    // Bobot layer hidden1 ke layer hidden2
    for (m = 1; m <= node_hidden2; m++)
    {
        for (n = 1; n <= node_hidden1; n++)
        {
            nilai_random = rand()%1000;
            nilai_random = ((nilai_random / 1000) - 0.5);
            WL2 [m][n] = nilai_random;
        }
    }
    // Bobot layer hidden2 ke layer output
    for (m = 1; m <= node_output; m++)
    {
        for (n = 1; n <= node_hidden2; n++)
        {
            nilai_random = rand()%1000;
            nilai_random = ((nilai_random / 1000) - 0.5);
            WL3 [m][n] = nilai_random;
        }
    }
    // Bobot layer hidden2 ke layer output
    for (m = 1; m <= node_output; m++)
    {
        for (n = 1; n <= node_hidden2; n++)
        {
            nilai_random = rand()%1000;
            nilai_random = ((nilai_random / 1000) - 0.5);
            WLOUT [m][n] = nilai_random;
        }
    }
}

// mengisi nilai bias pada layer hidden1 bernilai random

```



```

        for (n = 1; n <= node_hidden1; n++)
        {
            nilai_random = rand()%1000;
            nilai_random = ((nilai_random / 1000) - 0.5);
            BIAS [1][n] = nilai_random;
        }

// mengisi nilai bias pada layer hidden2 bernilai random
for (n = 1; n <= node_hidden2; n++)
{
    nilai_random = rand()%1000;
    nilai_random = ((nilai_random / 1000) - 0.5);
    BIAS [2][n] = nilai_random;
}

// mengisi nilai bias pada layer output bernilai random
for (n = 1; n <= node_output; n++)
{
    nilai_random = rand()%1000;
    nilai_random = ((nilai_random / 1000) - 0.5);
    BIAS [3][n] = nilai_random;
}
}
else
{
    // mengisi nilai semua bobot dan bias bernilai sama yaitu = 1
    // Bobot input ke layer hidden1
    for (m = 1; m <= node_hidden1; m++)
    {
        for (n = 1; n <= banyak_input; n++)
        {
            WL1 [m][n] = 1;
        }
    }

    // Bobot layer hidden1 ke layer hidden2
    for (m = 1; m <= node_hidden2; m++)
    {
        for (n = 1; n <= node_hidden1; n++)
        {
            WL2 [m][n] = 1;
        }
    }

    // Bobot layer hidden2 ke layer output
    for (m = 1; m <= node_output; m++)
    {
        for (n = 1; n <= node_hidden2; n++)
        {
            WL3 [m][n] = 1;
        }
    }

    // Bobot layer hidden2 ke layer output
    for (m = 1; m <= node_output; m++)
    {
        for (n = 1; n <= node_hidden2; n++)
        {
            WLOUT [m][n] = 1;
        }
    }

    // mengisi nilai bias pada layer hidden1 bernilai 1
    for (n = 1; n <= node_hidden1; n++)
    {
        BIAS [1][n] = 1;
    }

    // mengisi nilai bias pada layer hidden2 bernilai 1
    for (n = 1; n <= node_hidden2; n++)
    {
        BIAS [2][n] = 1;
    }

    // mengisi nilai bias pada layer output bernilai 1
    for (n = 1; n <= node_output; n++)
    {

```

```

        BIAS [3][n] = 1;
    }
}

//Button start learning
private: System::Void button20_Click(System::Object^ sender, System::EventArgs^ e) {
    int j,m,n;
    float temp;
    bool status_delay;
    iterasi=0;
    MSE=0;
    banyak_data = int::Parse(textBox1->Text);
    banyak_input = int::Parse(textBox2->Text);
    node_hidden1 = int::Parse(textBox3->Text);
    node_hidden2 = int::Parse(textBox37->Text);
    node_output = int::Parse(textBox4->Text);
    //=====
    miu_nol = float::Parse(textBox27->Text);
    alpha = float::Parse(textBox26->Text);
    beta = float::Parse(textBox38->Text);
    tao = float::Parse(textBox8->Text);
    polake=0;
    iterasi=0;
    //=====
    status_learning=true;
    status_delay=false;
    if (checkBox1->Checked==true){status_delay=true;};
    if (status_delay==false) // tanpa delay (tidak menggunakan timer)
    {
        while (status_learning==true)
        {
            iterasi = iterasi + 1;
            for (j = 0; j <= banyak_data; j++) // banyak patern data
            {
                // y = fungsi sigmoid(sigma w*x) untuk hidden layer ke-1
                for (m = 1; m <= node_hidden1; m++)
                {
                    temp=0;
                    for (n = 1; n <= banyak_input; n++)
                    {
                        temp = temp + (WL1[m][n] * XIN[j][n]);
                    }
                    temp = temp + BIAS [1][m];
                    HASIL_LAYER[1][m] = 1/(1+exp(-alpha * temp));
                    HASIL_G [1][m] = alpha*HASIL_LAYER[1][m] * (1-HASIL_LAYER[1][m]);
                    Y[1][m] = HASIL_LAYER[1][m];
                }
                // y = fungsi sigmoid(sigma w*x) untuk hidden layer ke-2
                for (m = 1; m <= node_hidden2; m++)
                {
                    temp = 0;
                    for (n = 1; n <= node_hidden1; n++)
                    {
                        temp = temp + (WL2[m][n] * Y[1][n]);
                    }
                    temp = temp + BIAS [2][m];
                    HASIL_LAYER[2][m] = 1/(1+exp(-alpha * temp));
                    HASIL_G [2][m] = alpha*HASIL_LAYER[2][m] * (1-HASIL_LAYER[2][m]);
                    Y[2][m] = HASIL_LAYER[2][m];
                }
            }
            // y = fungsi sigmoid(sigma w*x) untuk hidden2 layer output
            for (m = 1; m <= node_output; m++)
            {
                temp = 0;
                for (n = 1; n <= node_hidden2; n++)
                {
                    temp = temp + (WL3[m][n] * Y[2][n]);
                }
                temp = temp + BIAS [3][m];
            }
        }
    }
}

```

```

        HASIL_LAYER[3][m] = 1/(1+exp(-alpha * temp));
        HASIL_G [3][m] = alpha*HASIL_LAYER[3][m] * (1-HASIL_LAYER[3][m]);
        Y[3][m] = HASIL_LAYER[3][m];
        //HITUNG ERROR
        OUT_ERROR [j][m] = TARGET [j][m] - Y[3][m];
        ERROR_AKHIR = ERROR_AKHIR + (powf(OUT_ERROR [j][m],2))/2;
    }

    // Mencari nilai Delta Layer Output
    for (m = 1; m <= node_output; m++)
    {
        DELTA [3][m] = OUT_ERROR [j][m] * HASIL_G [3][m];
    }
    // Mencari nilai Delta Layer Hidden 2
    for (m = 1; m <= node_hidden2; m++)
    {
        temp = 0;
        for (n = 1; n <= node_output; n++)
        {
            temp = temp + (DELTA[3][n] * WL3[n][m]);
        }
        DELTA [2][m] = temp * HASIL_G [2][m];
    }
    // Mencari nilai Delta Layer Hidden 1
    for (m = 1; m <= node_hidden1; m++)
    {
        temp = 0;
        for (n = 1; n <= node_hidden2; n++)
        {
            temp = temp + (DELTA[2][n] * WL2[n][m]);
        }
        DELTA [1][m] = temp * HASIL_G [1][m];
    }
    miu = miu_nol;
    if (radioButton3->Checked==true) // Memilih update miu
    {
        miu = miu_nol/(1+iterasi/tao);
        if (miu < 0.01) {miu = 0.01;};
    }
    // Update Bobot Layer hidden ke layer output
    for (m = 1; m <= node_output; m++)
    {
        for (n = 1; n<= node_hidden2; n++)
        {
            WL3[m][n] = WL3[m][n] + (miu * DELTA [3][m] * HASIL_LAYER[2][n]);
        }
    }
    // Update Bobot Layer hidden1 ke layer hidden2
    for (m = 1; m <= node_hidden2; m++)
    {
        for (n = 1; n<= node_hidden1; n++)
        {
            WL2[m][n] = WL2[m][n] + (miu * DELTA [2][m] * HASIL_LAYER[1][n]);
        }
    }

    // Update Bobot Layer hidden
    for (m = 1; m <= node_hidden1; m++)
    {
        for (n = 1; n<= banyak_input; n++)
        {
            WL1[m][n] = WL1[m][n] + (miu * DELTA[1][m] * XIN[j][n]);
        }
    }
    // cek apakah pilih update bias
    if (radioButton4->Checked==true)
    {
        // update bias pada layer output
        for (m = 1; m<= node_output; m++)
        {
            BIAS [3][m] = alpha * DELTA [3][m];
        }
    }

```

```

    }
    // update bias pada layer output
    for (m = 1; m <= node_hidden2; m++)
    {
        BIAS [2][m] = alpha * DELTA [2][m];
    }
    // update bias pada layer hidden
    for (m = 1; m <= node_hidden1; m++)
    {
        BIAS [1][m] = alpha * DELTA [1][m];
    }
};
} // akhir perulangan banyak patern data

textBox28->Text = Convert::ToString(iterasi);
MSE = ERROR_AKHIR / (banyak_data+1);
textBox29->Text = Convert::ToString(MSE);
ERROR_AKHIR=0;
ENERGI=0;
if (MSE < beta) {status_learning=false;};
} // end while
};

if (status_delay==true){timer_JST->Enabled=true;}; // menggunakan timer
}

//Proses Learning dengan pilihan ada delay (menggunakan timer)
private: System::Void timer_JST_Tick(System::Object^ sender, System::EventArgs^ e)
{
    int j,m,n;
    float temp;
    iterasi = iterasi + 1;
    for (j = 0; j <= banyak_data; j++) // banyak patern data
    {
        // y = fungsi sigmoid(sigma w*x) untuk hidden layer ke-1
        for (m = 1; m <= node_hidden1; m++)
        {
            temp=0;
            for (n = 1; n <= banyak_input; n++)
            {
                temp = temp + (WL1[m][n] * XIN[j][n]);
            }
            temp = temp + BIAS [1][m];
            HASIL_LAYER[1][m] = 1/(1+exp(-alpha * temp));
            HASIL_G [1][m] = alpha*HASIL_LAYER[1][m] * (1-HASIL_LAYER[1][m]);
            Y[1][m] = HASIL_LAYER[1][m];
        }
        // y = fungsi sigmoid(sigma w*x) untuk hidden layer ke-2
        for (m = 1; m <= node_hidden2; m++)
        {
            temp = 0;
            for (n = 1; n <= node_hidden1; n++)
            {
                temp = temp + (WL2[m][n] * Y[1][n]);
            }
            temp = temp + BIAS [2][m];
            HASIL_LAYER[2][m] = 1/(1+exp(-alpha * temp));
            HASIL_G [2][m] = alpha*HASIL_LAYER[2][m] * (1-HASIL_LAYER[2][m]);
            Y[2][m] = HASIL_LAYER[2][m];
        }
        // y = fungsi sigmoid(sigma w*x) untuk hidden2 layer output
        for (m = 1; m <= node_output; m++)
        {
            temp = 0;
            for (n = 1; n <= node_hidden2; n++)
            {
                temp = temp + (WL3[m][n] * Y[2][n]);
            }
            temp = temp + BIAS [3][m];
            HASIL_LAYER[3][m] = 1/(1+exp(-alpha * temp));
            HASIL_G [3][m] = alpha*HASIL_LAYER[3][m] * (1-HASIL_LAYER[3][m]);
        }
    }
}

```

```

        Y[3][m] = HASIL_LAYER[3][m];
        //HITUNG ERROR
        OUT_ERROR [j][m] = TARGET [j][m] - Y[3][m];
        ERROR_AKHIR = ERROR_AKHIR + (powf(OUT_ERROR [j][m],2))/2;
    }

    // Mencari nilai Delta Layer Output
    for (m = 1; m <= node_output; m++)
    {
        DELTA [3][m] = OUT_ERROR [j][m] * HASIL_G [3][m];
    }
    // Mencari nilai Delta Layer Hidden 2
    for (m = 1; m <= node_hidden2; m++)
    {
        temp = 0;
        for (n = 1; n <= node_output; n++)
        {
            temp = temp + (DELTA[3][n] * WL3[n][m]);
        }
        DELTA [2][m] = temp * HASIL_G [2][m];
    }
    // Mencari nilai Delta Layer Hidden 1
    for (m = 1; m <= node_hidden1; m++)
    {
        temp = 0;
        for (n = 1; n <= node_hidden2; n++)
        {
            temp = temp + (DELTA[2][n] * WL2[n][m]);
        }
        DELTA [1][m] = temp * HASIL_G [1][m];
    }

    miu = miu_nol;
    if (radioButton3->Checked==true) // Memilih update miu
    {
        miu = miu_nol/(1+iterasi/tao);
        if (miu < 0.01) {miu = 0.01;};
    }
    // Update Bobot Layer hidden ke layer output
    for (m = 1; m <= node_output; m++)
    {
        for (n = 1; n<= node_hidden2; n++)
        {
            WL3[m][n] = WL3[m][n] + (miu * DELTA [3][m] * HASIL_LAYER[2][n]);
        }
    }
    // Update Bobot Layer hidden1 ke layer hidden2
    for (m = 1; m <= node_hidden2; m++)
    {
        for (n = 1; n<= node_hidden1; n++)
        {
            WL2[m][n] = WL2[m][n] + (miu * DELTA [2][m] * HASIL_LAYER[1][n]);
        }
    }
    // Update Bobot Layer hidden
    for (m = 1; m <= node_hidden1; m++)
    {
        for (n = 1; n<= banyak_input; n++)
        {
            WL1[m][n] = WL1[m][n] + (miu * DELTA[1][m] * XIN[j][n]);
        }
    }
    // cek apakah pilih update bias
    if (radioButton4->Checked==true)
    {
        // update bias pada layer output
        for (m = 1; m<= node_output; m++)
        {
            BIAS [3][m] = alpha * DELTA [3][m];
        }
        // update bias pada layer output
    }

```

```

        for (m = 1; m<= node_hidden2; m++)
        {
            BIAS [2][m] = alpha * DELTA [2][m];
        }
        // update bias pada layer hidden
        for (m = 1; m<= node_hidden1; m++)
        {
            BIAS [1][m] = alpha * DELTA [1][m];
        }
    };
} // akhir perulangan banyak patern data

textBox28->Text = Convert::ToString(iterasi);
MSE = ERROR_AKHIR / (banyak_data+1);
textBox29->Text = Convert::ToString(MSE);
ERROR_AKHIR=0;
if (MSE < beta) {timer_JST->Enabled=false;};
}

// Ambil data input dan target
private: System::Void button24_Click(System::Object^ sender, System::EventArgs^ e) {
    int kk=0,ll=0,ulang=0,m1=0,datanya=0;
    char ss;
    System::String^ bantu;
    StreamReader^ baca_file;
    StreamWriter^ tulis_file;
    button24->Enabled=false;
    //=====
    // Ambil data Input Learning
    baca_file = gcnew StreamReader("Data_INPUT_LEARNING_ANN.txt");
    tulis_file = gcnew StreamWriter ("Data INPUT 5 VARIASI.txt");
    tulis_file->Write("Hasil Baca Data INPUT");
    tulis_file->Write("\r\n");
    bantu = baca_file->ReadLine(); // baca keterangan atas
    for (kk=1; kk<=170; kk++)
    {
        tulis_file->Write(Convert::ToString(kk));
        tulis_file->Write(".: ");
        bantu = baca_file->ReadLine();
        for (m1=0; m1<=39; m1++)
        {
            ss=bantu[m1];
            if (ss==48) {datanya=0;}else{datanya=1;} // 48 nilai ascii angka 0 dan 49
            nilai ascii angka 1
            XIN[kk][m1] = datanya;
            tulis_file->Write(Convert::ToString(XIN[kk][m1]));
        }
        ss=bantu[40]; // karakter # pagar
        tulis_file->Write("#");
        // Ambil data target
        for (m1=1; m1<=6; m1++)
        {
            ss=bantu[40+m1];
            if (ss==48) {datanya=0;}else{datanya=1;}
            TARGET[kk][m1] = datanya;
            tulis_file->Write(Convert::ToString(TARGET[kk][m1]));
        }
        tulis_file->Write("\r\n");
    }
    baca_file->Close();
    tulis_file->Close();
    cvWaitKey(50);
    button24->Enabled=true;
} // akhir proses ambil data input dan target

// STOP learning
private: System::Void button21_Click(System::Object^ sender, System::EventArgs^ e) {
    timer_JST->Enabled=false;
    status_learning=false;
}

```

```

// button tes training / uji coba hasil learning
private: System::Void button23_Click(System::Object^ sender, System::EventArgs^ e) {
int m=0,n=0,bit0=0,bit1=0,bit2=0,bit3=0,bit4=0,bit5=0,jumlahdata=0;
double thres=0;
// proses ANN
for (m = 1; m <= node_hidden1; m++)
{
temp=0;
for (n = 1; n <= banyak_input; n++)
{
temp = temp + (WL1[m][n] * X_TRAIN[n]);
}
temp = temp + BIAS [1][m];
HASIL_LAYER[1][m] = 1/(1+exp(-alpha * temp));
HASIL_G [1][m] = alpha*HASIL_LAYER[1][m] * (1-HASIL_LAYER[1][m]);
Y[1][m] = HASIL_LAYER[1][m];
}

// y = fungsi sigmoid(sigma w*x) untuk hidden layer ke-2
for (m = 1; m <= node_hidden2; m++)
{
temp = 0;
for (n = 1; n <= node_hidden1; n++)
{
temp = temp + (WL2[m][n] * Y[1][n]);
}
temp = temp + BIAS [2][m];
HASIL_LAYER[2][m] = 1/(1+exp(-alpha * temp));
HASIL_G [2][m] = alpha*HASIL_LAYER[2][m] * (1-HASIL_LAYER[2][m]);
Y[2][m] = HASIL_LAYER[2][m];
}

// y = fungsi sigmoid(sigma w*x) untuk hidden2 layer output
for (m = 1; m <= node_output; m++)
{
temp = 0;
for (n = 1; n <= node_hidden2; n++)
{
temp = temp + (WL3[m][n] * Y[2][n]);
}
temp = temp + BIAS [3][m];
HASIL_LAYER[3][m] = 1/(1+exp(-alpha * temp));
HASIL_G [3][m] = alpha*HASIL_LAYER[3][m] * (1-HASIL_LAYER[3][m]);
Y[3][m] = HASIL_LAYER[3][m];
}

//=====
textBox31->Text=Convert::ToString(Y[3][1]);
textBox32->Text=Convert::ToString(Y[3][2]);
textBox33->Text=Convert::ToString(Y[3][3]);
textBox34->Text=Convert::ToString(Y[3][4]);
textBox35->Text=Convert::ToString(Y[3][5]);
textBox6->Text=Convert::ToString(Y[3][6]);
//=====
thres = double::Parse(textBox36->Text);
if (Y[3][1] >= thres){bit0=1;};
if (Y[3][2] >= thres){bit1=1;};
if (Y[3][3] >= thres){bit2=1;};
if (Y[3][4] >= thres){bit3=1;};
if (Y[3][5] >= thres){bit4=1;};
if (Y[3][6] >= thres){bit5=1;};
textBox49->Text=Convert::ToString(bit0);
textBox48->Text=Convert::ToString(bit1);
textBox47->Text=Convert::ToString(bit2);
textBox46->Text=Convert::ToString(bit3);
textBox45->Text=Convert::ToString(bit4);
textBox7->Text=Convert::ToString(bit5);
//=====
jumlahdata=0;
if (bit0==1){jumlahdata = jumlahdata + 1;};
if (bit1==1){jumlahdata = jumlahdata + 2;};

```

```

if (bit2==1){jumlahdata = jumlahdata + 4;};
if (bit3==1){jumlahdata = jumlahdata + 8;};
if (bit4==1){jumlahdata = jumlahdata + 16;};
if (bit5==1){jumlahdata = jumlahdata + 32;};
//=====
if (jumlahdata==0){label66->Text="Spasi";};
if (jumlahdata==1){label66->Text="A";};
if (jumlahdata==2){label66->Text="B";};
if (jumlahdata==3){label66->Text="C";};
if (jumlahdata==4){label66->Text="D";};
if (jumlahdata==5){label66->Text="E";};
if (jumlahdata==6){label66->Text="F";};
if (jumlahdata==7){label66->Text="G";};
if (jumlahdata==8){label66->Text="H";};
if (jumlahdata==9){label66->Text="I";};
if (jumlahdata==10){label66->Text="J";};
if (jumlahdata==11){label66->Text="K";};
if (jumlahdata==12){label66->Text="L";};
if (jumlahdata==13){label66->Text="M";};
if (jumlahdata==14){label66->Text="N";};
if (jumlahdata==15){label66->Text="O";};
if (jumlahdata==16){label66->Text="P";};
if (jumlahdata==17){label66->Text="Q";};
if (jumlahdata==18){label66->Text="R";};
if (jumlahdata==19){label66->Text="S";};
if (jumlahdata==20){label66->Text="T";};
if (jumlahdata==21){label66->Text="U";};
if (jumlahdata==22){label66->Text="V";};
if (jumlahdata==23){label66->Text="W";};
if (jumlahdata==24){label66->Text="X";};
if (jumlahdata==25){label66->Text="Y";};
if (jumlahdata==26){label66->Text="Z";};
if (jumlahdata==27){label66->Text="Kapital";};
if (jumlahdata==28){label66->Text="Angka";};
if (jumlahdata==29){label66->Text="Titik";};
if (jumlahdata==30){label66->Text="Tanya";};
if (jumlahdata==31){label66->Text="Koma";};
if (jumlahdata==32){label66->Text="! SERU";};
if (jumlahdata==33){label66->Text="- STRIP";};
}

// Ambil Bobot Akhir
private: System::Void button25_Click(System::Object^ sender, System::EventArgs^ e) {
if(openFileDialog1->ShowDialog() == System::Windows::Forms::DialogResult::OK)
{
    int jjj=0, kkk=0;
    System::String^ bantu;
    StreamReader^ baca_file = gcnew StreamReader(openFileDialog1->FileName);
    StreamWriter^ tulis_file = gcnew StreamWriter ("HASIL_BACA_BOBOT.txt");
    //=====
// Urutan: Iterasi, Beta, Alpha, node Input, Node Hidden1, Node Hidden2, Node Output,
// Bobot L1 L2 L3, dan Bias 1 2 3
    bantu = baca_file->ReadLine(); // keterangan atas
    tulis_file->Write(bantu);
    tulis_file->Write("\r\n");
    bantu = baca_file->ReadLine(); // keterangan urutan
    tulis_file->Write(bantu);
    tulis_file->Write("\r\n");
    bantu = baca_file->ReadLine(); // iterasi
    tulis_file->Write(bantu);
    tulis_file->Write("\r\n");
    bantu = baca_file->ReadLine(); // beta
    tulis_file->Write(bantu);
    tulis_file->Write("\r\n");
    bantu = baca_file->ReadLine(); // alpha
    alpha = double::Parse(bantu);
    tulis_file->Write(bantu);
    tulis_file->Write("\r\n");
    bantu = baca_file->ReadLine(); // data input
    banyak_input = int::Parse(bantu);
}
}

```



```

tulis_file->Write(Convert::ToString(banyak_input));
tulis_file->Write("\r\n");
bantu = baca_file->ReadLine(); // node hidden1
node_hidden1 = int::Parse(bantu);
tulis_file->Write(Convert::ToString(node_hidden1));
tulis_file->Write("\r\n");
bantu = baca_file->ReadLine(); // node hidden2
node_hidden2 = int::Parse(bantu);
tulis_file->Write(Convert::ToString(node_hidden2));
tulis_file->Write("\r\n");
bantu = baca_file->ReadLine(); // node output
node_output = int::Parse(bantu);
tulis_file->Write(Convert::ToString(node_output));
tulis_file->Write("\r\n");
bantu = baca_file->ReadLine(); // keterangan ==== Weight layer 1 =====
tulis_file->Write(bantu);
tulis_file->Write("\r\n");
//=====
for (jjj=1; jjj<=node_hidden1; jjj++)
{
    for (kkk=1; kkk<=banyak_input; kkk++)
    {
        bantu = baca_file->ReadLine(); // ambil bobot
        WL1[jjj][kkk] = double::Parse(bantu);
        tulis_file->Write(Convert::ToString(WL1[jjj][kkk]));
        tulis_file->Write("\r\n");
    }
}
bantu = baca_file->ReadLine(); // keterangan ==== Weight layer 2 =====
tulis_file->Write(bantu);
tulis_file->Write("\r\n");
//=====
for (jjj=1; jjj<=node_hidden2; jjj++)
{
    for (kkk=1; kkk<=node_hidden1; kkk++)
    {
        bantu = baca_file->ReadLine(); // ambil bobot
        WL2[jjj][kkk] = double::Parse(bantu);
        tulis_file->Write(Convert::ToString(WL2[jjj][kkk]));
        tulis_file->Write("\r\n");
    }
}
bantu = baca_file->ReadLine(); // keterangan ==== Weight layer 3 =====
tulis_file->Write(bantu);
tulis_file->Write("\r\n");
//=====
for (jjj=1; jjj<=node_output; jjj++)
{
    for (kkk=1; kkk<=node_hidden2; kkk++)
    {
        bantu = baca_file->ReadLine(); // ambil bobot
        WL3[jjj][kkk] = double::Parse(bantu);
        tulis_file->Write(Convert::ToString(WL3[jjj][kkk]));
        tulis_file->Write("\r\n");
    }
}
bantu = baca_file->ReadLine(); // keterangan ==== Bias layer 1 =====
tulis_file->Write(bantu);
tulis_file->Write("\r\n");
//=====
for (jjj=1; jjj<=node_hidden1; jjj++)
{
    bantu = baca_file->ReadLine(); // ambil BIAS 1
    BIAS [1][jjj] = double::Parse(bantu);
    tulis_file->Write(Convert::ToString(BIAS [1][jjj]));
    tulis_file->Write("\r\n");
}
bantu = baca_file->ReadLine(); // keterangan ==== Bias layer 2 =====
tulis_file->Write(bantu);
tulis_file->Write("\r\n");
//=====

```

```

for (jjj=1; jjj<=node_hidden2; jjj++)
{
    bantu = baca_file->ReadLine(); // ambil BIAS 2
    BIAS [2][jjj] = double::Parse(bantu);
    tulis_file->Write(Convert::ToString(BIAS [2][jjj]));
    tulis_file->Write("\r\n");
}
bantu = baca_file->ReadLine(); // keterangan ==== Bias layer 3 =====
tulis_file->Write(bantu);
tulis_file->Write("\r\n");
//=====
for (jjj=1; jjj<=node_output; jjj++)
{
    bantu = baca_file->ReadLine(); // ambil BIAS 3
    BIAS [3][jjj] = double::Parse(bantu);
    tulis_file->Write(Convert::ToString(BIAS [3][jjj]));
    tulis_file->Write("\r\n");
}
//=====
baca_file->Close();
tulis_file->Close();
MessageBox::Show("Nilai Bobot dan Bias sudah diambil !", "Braille Reader - JST",
    MessageBoxButtons::OK, MessageBoxIcon::Asterisk);
}

}

// Simpan Bobot dan bias akhir
private: System::Void button18_Click(System::Object^ sender, System::EventArgs^ e) {
int mm=0,nn=0;
//=====
StreamWriter^ tulis_file = gcnew StreamWriter (textBox25->Text+".txt");
tulis_file->Write("Hasil Bobot dan Bias Akhir");
tulis_file->Write("\r\n"); // Enter ganti baris
tulis_file->Write("Urutan: Iterasi, Beta, Alpha, node Input, Node Hidden1, Node Hidden2,
Node Output, Bobot L1 L2 L3, dan Bias 1 2 3 ");
tulis_file->Write("\r\n");
tulis_file->Write(Convert::ToString(iterasi));
tulis_file->Write("\r\n");
tulis_file->Write(Convert::ToString(beta));
tulis_file->Write("\r\n");
tulis_file->Write(Convert::ToString(alpha));
tulis_file->Write("\r\n");
tulis_file->Write(Convert::ToString(banyak_input));
tulis_file->Write("\r\n");
tulis_file->Write(Convert::ToString(node_hidden1));
tulis_file->Write("\r\n");
tulis_file->Write(Convert::ToString(node_hidden2));
tulis_file->Write("\r\n");
tulis_file->Write(Convert::ToString(node_output));
    tulis_file->Write("\r\n");
    tulis_file->Write("==== Bobot Layer 1 =====");
    tulis_file->Write("\r\n");
    //=====
for (mm=1; mm<=node_hidden1; mm++)
{
    for (nn=1; nn<=banyak_input; nn++)
    {
        tulis_file->Write(Convert::ToString(WL1[mm][nn]));
        tulis_file->Write("\r\n");
    }
}
tulis_file->Write("==== Bobot Layer 2 =====");
tulis_file->Write("\r\n");
//=====
for (mm=1; mm<=node_hidden2; mm++)
{
    for (nn=1; nn<=node_hidden1; nn++)
    {
        tulis_file->Write(Convert::ToString(WL2[mm][nn]));
        tulis_file->Write("\r\n");
    }
}
}

```

```

    }
}
tulis_file->Write("==== Bobot Layer 3 ===");
tulis_file->Write("\r\n");
//=====
for (mm=1; mm<=node_output; mm++)
{
    for (nn=1; nn<=node_hidden2; nn++)
    {
        tulis_file->Write(Convert::ToString(WL3[mm][nn]));
        tulis_file->Write("\r\n");
    }
}
tulis_file->Write("==== Bias Layer 1 ===");
tulis_file->Write("\r\n");
//=====
for (nn=1; nn<=node_hidden1; nn++)
{
    tulis_file->Write(Convert::ToString(BIAS[1][nn]));
    tulis_file->Write("\r\n");
}
tulis_file->Write("==== Bias Layer 2 ===");
tulis_file->Write("\r\n");
//=====
for (nn=1; nn<=node_hidden2; nn++)
{
    tulis_file->Write(Convert::ToString(BIAS[2][nn]));
    tulis_file->Write("\r\n");
}
tulis_file->Write("==== Bias Layer 3 ===");
tulis_file->Write("\r\n");
//=====
for (nn=1; nn<=node_output; nn++)
{
    tulis_file->Write(Convert::ToString(BIAS[3][nn]));
    tulis_file->Write("\r\n");
}
tulis_file->Close();
//=====
    MessageBox::Show("Nilai Bobot dan bias sudah disimpan !", "Braille Reader - JST",
    MessageBoxButtons::OK, MessageBoxIcon::Asterisk);
}

```

```

// RS-232 TERIMA Data serial
private: System::Void serialPort1_DataReceived(System::Object^ sender,
System::IO::Ports::SerialDataReceivedEventArgs^ e) {
    terima = serialPort1->ReadExisting();
    this->Invoke(gcnew EventHandler(this, &Form1::SetTextCallback));
}

// button luruskan otomatis
private: System::Void btn_luruskan_auto_Click(System::Object^ sender, System::EventArgs^
e) {
    float sudut135 = 1.5;
    float sudut246 = -0.8;
    //===== Proses luruskan image 1
    if (origimage1 != 0)
    {
        if(hasil_putar1 == 0)
        {
            hasil_putar1 = cvCreateImage(cvSize(origimage1->width, origimage1->height),
origimage1->depth, 1);
        }
        if(hasil_putar1 != 0)
        {

```

```

        cvReleaseImage( &hasil_putar1 ); // fungsinya menghapus image jika tidak digunakan
        hasil_putar1 = cvCreateImage(cvSize(origimage1->width, origimage1->height),
origimage1->depth, 1);
    }
    hasil_putar1=cvCloneImage(origimage1);
    //=====
    float m[6];
    CvMat M = cvMat(2, 3, CV_32F, m);
    int w = hasil_putar1->width;
    int h = hasil_putar1->height;
    float angleRadians = sudut135 * ((float)CV_PI / 180.0f);
    m[0] = (float)( cos(angleRadians) );
    m[1] = (float)( sin(angleRadians) );
    m[3] = -m[1];
    m[4] = m[0];
    m[2] = w*0.5f;
    m[5] = h*0.5f;
    // Make a spare image for the result
    CvSize sizeRotated;
    sizeRotated.width = cvRound(w);
    sizeRotated.height = cvRound(h);
    // Rotate
    IplImage *imageRotated = cvCreateImage( sizeRotated, hasil_putar1->depth, hasil_putar1-
>nChannels );
    // Transform the image
    cvGetQuadrangleSubPix( hasil_putar1, imageRotated, &M);
    //=== Ambil hasil putar image 1
    if(hasil_putar1 == 0)
    {
        hasil_putar1 = cvCreateImage(cvSize(imageRotated->width, imageRotated->height),
imageRotated->depth, 1);
    }
    if(hasil_putar1 != imageRotated)
    {
        cvReleaseImage( &hasil_putar1 ); // fungsinya menghapus image jika tidak digunakan
        hasil_putar1 = cvCreateImage(cvSize(imageRotated->width, imageRotated->height),
imageRotated->depth, 1);
    }
    hasil_putar1=cvCloneImage(imageRotated);
    //=== Isi image tampung 1 ===
    if(tampung_1 == 0)
    {
        tampung_1 = cvCreateImage(cvSize(hasil_putar1->width, hasil_putar1->height), hasil_putar1-
>depth, 1);
    }

    if(tampung_1 != 0)
    {
        cvReleaseImage( &tampung_1 ); // fungsinya menghapus image jika tidak digunakan
        tampung_1 = cvCreateImage(cvSize(hasil_putar1->width, hasil_putar1->height), hasil_putar1-
>depth, 1);
    }
    tampung_1=cvCloneImage(hasil_putar1);
    //=====
    };
    //=====
    //===== Proses luruskan image 2
    if (origimage2 != 0)
    {
        if(hasil_putar2 == 0)
        {
            hasil_putar2 = cvCreateImage(cvSize(origimage2->width, origimage2->height),
origimage2->depth, 1);
        }
        if(hasil_putar2 != 0)
        {
            cvReleaseImage( &hasil_putar2 ); // fungsinya menghapus image jika tidak digunakan
            hasil_putar2 = cvCreateImage(cvSize(origimage2->width, origimage2->height),
origimage2->depth, 1);

```

```

    }
    hasil_putar2=cvCloneImage(origimage2);
    //=====
    float m[6];
    CvMat M = cvMat(2, 3, CV_32F, m);
    int w = hasil_putar2->width;
    int h = hasil_putar2->height;
    float angleRadians = sudut135 * ((float)CV_PI / 180.0f);
    m[0] = (float)( cos(angleRadians) );
    m[1] = (float)( sin(angleRadians) );
    m[3] = -m[1];
    m[4] = m[0];
    m[2] = w*0.5f;
    m[5] = h*0.5f;
    // Make a spare image for the result
    CvSize sizeRotated;
    sizeRotated.width = cvRound(w);
    sizeRotated.height = cvRound(h);
    // Rotate
    IplImage *imageRotated = cvCreateImage( sizeRotated, hasil_putar2->depth, hasil_putar2->nChannels );
    // Transform the image
    cvGetQuadrangleSubPix( hasil_putar2, imageRotated, &M);
    //=== Ambil hasil putar image 2
    if(hasil_putar2 == 0)
    {
        hasil_putar2 = cvCreateImage(cvSize(imageRotated->width, imageRotated->height),
        imageRotated->depth, 1);
    }
    if(hasil_putar2 != imageRotated)
    {
        cvReleaseImage( &hasil_putar2 ); // fungsinya menghapus image jika tidak digunakan
        hasil_putar2 = cvCreateImage(cvSize(imageRotated->width, imageRotated->height),
        imageRotated->depth, 1);
    }
    hasil_putar2=cvCloneImage(imageRotated);
    //=== Isi image tampung 2 ===
    if(tampung_2 == 0)
    {
        tampung_2 = cvCreateImage(cvSize(hasil_putar2->width, hasil_putar2->height), hasil_putar2->depth, 1);
    }

    if(tampung_2 != 0)
    {
        cvReleaseImage( &tampung_2 ); // fungsinya menghapus image jika tidak digunakan
        tampung_2 = cvCreateImage(cvSize(hasil_putar2->width, hasil_putar2->height), hasil_putar2->depth, 1);
    }
    tampung_2=cvCloneImage(hasil_putar2);
    //=====
    };
    //===== Proses luruskan image 3
    if (origimage3 != 0)
    {
        if(hasil_putar3 == 0)
        {
            hasil_putar3 = cvCreateImage(cvSize(origimage3->width, origimage3->height),
            origimage3->depth, 1);
        }
        if(hasil_putar3 != 0)
        {
            cvReleaseImage( &hasil_putar3 ); // fungsinya menghapus image jika tidak digunakan
            hasil_putar3 = cvCreateImage(cvSize(origimage3->width, origimage3->height),
            origimage3->depth, 1);
        }
        hasil_putar3=cvCloneImage(origimage3);
        //=====
        float m[6];
        CvMat M = cvMat(2, 3, CV_32F, m);
        int w = hasil_putar3->width;

```

```

int h = hasil_putar3->height;
float angleRadians = sudut135 * ((float)CV_PI / 180.0f);
m[0] = (float)( cos(angleRadians) );
m[1] = (float)( sin(angleRadians) );
m[3] = -m[1];
m[4] = m[0];
m[2] = w*0.5f;
m[5] = h*0.5f;
// Make a spare image for the result
CvSize sizeRotated;
sizeRotated.width = cvRound(w);
sizeRotated.height = cvRound(h);
// Rotate
IplImage *imageRotated = cvCreateImage( sizeRotated, hasil_putar3->depth, hasil_putar3->nChannels );
// Transform the image
cvGetQuadrangleSubPix( hasil_putar3, imageRotated, &M);
//=== Ambil hasil putar image 3
if(hasil_putar3 == 0)
{
    hasil_putar3 = cvCreateImage(cvSize(imageRotated->width, imageRotated->height),
imageRotated->depth, 1);
}
if(hasil_putar3 != imageRotated)
{
    cvReleaseImage( &hasil_putar3 ); // fungsinya menghapus image jika tidak digunakan
    hasil_putar3 = cvCreateImage(cvSize(imageRotated->width, imageRotated->height),
imageRotated->depth, 1);
}
hasil_putar3=cvCloneImage(imageRotated);
//=== Isi image tampung 3 ===
if(tampung_3 == 0)
{
    tampung_3 = cvCreateImage(cvSize(hasil_putar3->width, hasil_putar3->height), hasil_putar3->depth, 1);
}

if(tampung_3 != 0)
{
    cvReleaseImage( &tampung_3 ); // fungsinya menghapus image jika tidak digunakan
    tampung_3 = cvCreateImage(cvSize(hasil_putar3->width, hasil_putar3->height), hasil_putar3->depth, 1);
}
tampung_3=cvCloneImage(hasil_putar3);
//=====
};
//===== Proses luruskan image 4
if (origimage4 != 0)
{
    if(hasil_putar4 == 0)
    {
        hasil_putar4 = cvCreateImage(cvSize(origimage4->width, origimage4->height),
origimage4->depth, 1);
    }
    if(hasil_putar4 != 0)
    {
        cvReleaseImage( &hasil_putar4 ); // fungsinya menghapus image jika tidak digunakan
        hasil_putar4 = cvCreateImage(cvSize(origimage4->width, origimage4->height),
origimage4->depth, 1);
    }
    hasil_putar4=cvCloneImage(origimage4);
    //=====
    float m[6];
    CvMat M = cvMat(2, 3, CV_32F, m);
    int w = hasil_putar4->width;
    int h = hasil_putar4->height;
    float angleRadians = sudut135 * ((float)CV_PI / 180.0f);
    m[0] = (float)( cos(angleRadians) );
    m[1] = (float)( sin(angleRadians) );

```

```

m[3] = -m[1];
m[4] = m[0];
m[2] = w*0.5f;
m[5] = h*0.5f;
// Make a spare image for the result
CvSize sizeRotated;
sizeRotated.width = cvRound(w);
sizeRotated.height = cvRound(h);
// Rotate
IplImage *imageRotated = cvCreateImage( sizeRotated, hasil_putar4->depth, hasil_putar4->nChannels );
// Transform the image
cvGetQuadrangleSubPix( hasil_putar4, imageRotated, &M);
//=== Ambil hasil putar image 4
if(hasil_putar4 == 0)
{
    hasil_putar4 = cvCreateImage(cvSize(imageRotated->width, imageRotated->height),
imageRotated->depth, 1);
}
if(hasil_putar4 != imageRotated)
{
    cvReleaseImage( &hasil_putar4 ); // fungsinya menghapus image jika tidak digunakan
    hasil_putar4 = cvCreateImage(cvSize(imageRotated->width, imageRotated->height),
imageRotated->depth, 1);
}
hasil_putar4=cvCloneImage(imageRotated);
//=== Isi image tampung 4 ===
if(tampung_4 == 0)
{
    tampung_4 = cvCreateImage(cvSize(hasil_putar4->width, hasil_putar4->height), hasil_putar4->depth, 1);
}

if(tampung_4 != 0)
{
    cvReleaseImage( &tampung_4 ); // fungsinya menghapus image jika tidak digunakan
    tampung_4 = cvCreateImage(cvSize(hasil_putar4->width, hasil_putar4->height), hasil_putar4->depth, 1);
}
tampung_4=cvCloneImage(hasil_putar4);
//=====
};
//===== Proses luruskan image 5
if (origimage5 != 0)
{
    if(hasil_putar5 == 0)
    {
        hasil_putar5 = cvCreateImage(cvSize(origimage5->width, origimage5->height),
origimage5->depth, 1);
    }
    if(hasil_putar5 != 0)
    {
        cvReleaseImage( &hasil_putar5 ); // fungsinya menghapus image jika tidak digunakan
        hasil_putar5 = cvCreateImage(cvSize(origimage5->width, origimage5->height),
origimage5->depth, 1);
    }
    hasil_putar5=cvCloneImage(origimage5);
    //=====
    float m[6];
    CvMat M = cvMat(2, 3, CV_32F, m);
    int w = hasil_putar5->width;
    int h = hasil_putar5->height;
    float angleRadians = sudut135 * ((float)CV_PI / 180.0f);
    m[0] = (float)( cos(angleRadians) );
    m[1] = (float)( sin(angleRadians) );
    m[3] = -m[1];
    m[4] = m[0];
    m[2] = w*0.5f;
    m[5] = h*0.5f;

```

```

// Make a spare image for the result
CvSize sizeRotated;
sizeRotated.width = cvRound(w);
sizeRotated.height = cvRound(h);
// Rotate
IplImage *imageRotated = cvCreateImage( sizeRotated, hasil_putar5->depth, hasil_putar5->nChannels );
// Transform the image
cvGetQuadrangleSubPix( hasil_putar5, imageRotated, &M);
//== Ambil hasil putar image 5
if(hasil_putar5 == 0)
{
    hasil_putar5 = cvCreateImage(cvSize(imageRotated->width, imageRotated->height),
imageRotated->depth, 1);
}
if(hasil_putar5 != imageRotated)
{
    cvReleaseImage( &hasil_putar5 ); // fungsinya menghapus image jika tidak digunakan
    hasil_putar5 = cvCreateImage(cvSize(imageRotated->width, imageRotated->height),
imageRotated->depth, 1);
}
hasil_putar5=cvCloneImage(imageRotated);
//== Isi image tampung 5 ====
if(tampung_5 == 0)
{
    tampung_5 = cvCreateImage(cvSize(hasil_putar5->width, hasil_putar5->height), hasil_putar5->depth, 1);
}

if(tampung_5 != 0)
{
    cvReleaseImage( &tampung_5 ); // fungsinya menghapus image jika tidak digunakan
    tampung_5 = cvCreateImage(cvSize(hasil_putar5->width, hasil_putar5->height), hasil_putar5->depth, 1);
}
tampung_5=cvCloneImage(hasil_putar5);
//=====
};
//===== Proses luruskan image 6
if (origimage6 != 0)
{
    if(hasil_putar6 == 0)
    {
        hasil_putar6 = cvCreateImage(cvSize(origimage6->width, origimage6->height),
origimage6->depth, 1);
    }
    if(hasil_putar6 != 0)
    {
        cvReleaseImage( &hasil_putar6 ); // fungsinya menghapus image jika tidak digunakan
        hasil_putar6 = cvCreateImage(cvSize(origimage6->width, origimage6->height),
origimage6->depth, 1);
    }
    hasil_putar6=cvCloneImage(origimage6);
    //=====
    float m[6];
    CvMat M = cvMat(2, 3, CV_32F, m);
    int w = hasil_putar6->width;
    int h = hasil_putar6->height;
    float angleRadians = sudut135 * ((float)CV_PI / 180.0f);
    m[0] = (float)( cos(angleRadians) );
    m[1] = (float)( sin(angleRadians) );
    m[3] = -m[1];
    m[4] = m[0];
    m[2] = w*0.5f;
    m[5] = h*0.5f;
    // Make a spare image for the result
    CvSize sizeRotated;
    sizeRotated.width = cvRound(w);
    sizeRotated.height = cvRound(h);

```



```

// Rotate
IplImage *imageRotated = cvCreateImage( sizeRotated, hasil_putar6->depth, hasil_putar6-
>nChannels );
// Transform the image
cvGetQuadrangleSubPix( hasil_putar6, imageRotated, &M);
//=== Ambil hasil putar image 6
if(hasil_putar6 == 0)
{
    hasil_putar6 = cvCreateImage(cvSize(imageRotated->width, imageRotated->height),
imageRotated->depth, 1);
}
if(hasil_putar6 != imageRotated)
{
    cvReleaseImage( &hasil_putar6 ); // fungsinya menghapus image jika tidak digunakan
    hasil_putar6 = cvCreateImage(cvSize(imageRotated->width, imageRotated->height),
imageRotated->depth, 1);
}
hasil_putar6=cvCloneImage(imageRotated);
//=== Isi image tampung 6 ===
if(tampung_6 == 0)
{
    tampung_6 = cvCreateImage(cvSize(hasil_putar6->width, hasil_putar6->height), hasil_putar6-
>depth, 1);
}

if(tampung_6 != 0)
{
    cvReleaseImage( &tampung_6 ); // fungsinya menghapus image jika tidak digunakan
    tampung_6 = cvCreateImage(cvSize(hasil_putar6->width, hasil_putar6->height), hasil_putar6-
>depth, 1);
}
tampung_6=cvCloneImage(hasil_putar6);
//=====
};
//=====
};
if (pilih_operasi==0){this->button_tampilkan_RGB_Click_1(sender, e);}
if (pilih_operasi==1){this->btn_crop_auto_Click(sender, e);}
}

// button crop image
private: System::Void btn_crop_auto_Click(System::Object^ sender, System::EventArgs^ e)
{
    //=== Proses Crop image 1
    if (hasil_putar1 != 0)
    {
        if(hasil_crop1 == 0)
        {
            hasil_crop1 = cvCreateImage(cvSize(hasil_putar1->width,hasil_putar1->height),
hasil_putar1->depth, 1);
        }
        if(hasil_crop1 != 0)
        {
            cvReleaseImage( &hasil_crop1 ); // fungsinya menghapus image jika tidak digunakan
            hasil_crop1 = cvCreateImage(cvSize(hasil_putar1->width, hasil_putar1->height),
hasil_putar1->depth, 1);
        }
        hasil_crop1=cvCloneImage(hasil_putar1);
        //===== data koordinat area crop image 1
        int YY_awal = 40;
        int YY_akhir = 400;
        int XX_awal = 50;
        int XX_akhir = 505;
        //=====
        cvSetImageROI(hasil_crop1, cvRect(XX_awal,YY_awal,XX_akhir,YY_akhir));
        IplImage *tmp = cvCreateImage(cvGetSize(hasil_crop1),hasil_crop1->depth,hasil_crop1-
>nChannels);
        cvCopy(hasil_crop1, tmp, NULL);
        cvResetImageROI(hasil_crop1);
    }
}

```

```

hasil_crop1 = cvCloneImage(tmp);
//=== Isi image tampung 1 ===
if(tampung_1 == 0)
{
tampung_1 = cvCreateImage(cvSize(hasil_crop1->width, hasil_crop1->height), hasil_crop1->depth, 1);
}
if(tampung_1 != 0)
{
cvReleaseImage( &tampung_1 ); // fungsinya menghapus image jika tidak digunakan
tampung_1 = cvCreateImage(cvSize(hasil_crop1->width, hasil_crop1->height), hasil_crop1->depth, 1);
}
tampung_1=cvCloneImage(hasil_crop1);
//=====
};
//=====
//== Proses Crop image 2
if (hasil_putar2 != 0)
{
    if(hasil_crop2 == 0)
    {
        hasil_crop2 = cvCreateImage(cvSize(hasil_putar2->width,hasil_putar2->height),
hasil_putar2->depth, 1);
    }
    if(hasil_crop2 != 0)
    {
        cvReleaseImage( &hasil_crop2 ); // fungsinya menghapus image jika tidak digunakan
        hasil_crop2 = cvCreateImage(cvSize(hasil_putar2->width, hasil_putar2->height),
hasil_putar2->depth, 1);
    }
    hasil_crop2=cvCloneImage(hasil_putar2);
    //===== data koordinat area crop image 2
    int YY_awal = 15;
    int YY_akhir = 455;
    int XX_awal = 85;
    int XX_akhir = 550;
    //=====
    cvSetImageROI(hasil_crop2, cvRect(XX_awal,YY_awal,XX_akhir,YY_akhir));
    IplImage *tmp = cvCreateImage(cvGetSize(hasil_crop2),hasil_crop2->depth,hasil_crop2->nChannels);
    cvCopy(hasil_crop2, tmp, NULL);
    cvResetImageROI(hasil_crop2);
    hasil_crop2 = cvCloneImage(tmp);
    //== Isi image tampung 2 ==
    if(tampung_2 == 0)
    {
        tampung_2 = cvCreateImage(cvSize(hasil_crop2->width, hasil_crop2->height), hasil_crop2->depth, 1);
    }
    if(tampung_2 != 0)
    {
        cvReleaseImage( &tampung_2 ); // fungsinya menghapus image jika tidak digunakan
        tampung_2 = cvCreateImage(cvSize(hasil_crop2->width, hasil_crop2->height), hasil_crop2->depth, 1);
    }
    tampung_2=cvCloneImage(hasil_crop2);
    //=====
    };
    //=====
    //== Proses Crop image 3
    if (hasil_putar3 != 0)
    {
        if(hasil_crop3 == 0)
        {
            hasil_crop3 = cvCreateImage(cvSize(hasil_putar3->width,hasil_putar3->height),
hasil_putar3->depth, 1);
        }
        if(hasil_crop3 != 0)
        {
            cvReleaseImage( &hasil_crop3 ); // fungsinya menghapus image jika tidak digunakan

```

```

        hasil_crop3 = cvCreateImage(cvSize(hasil_putar3->width, hasil_putar3->height),
hasil_putar3->depth, 1);
    }
    hasil_crop3=cvCloneImage(hasil_putar3);
    //===== data koordinat area crop image 3
    int YY_awal = 2;
    int YY_akhir = 450;
    int XX_awal = 50;
    int XX_akhir = 508;
    //=====
    cvSetImageROI(hasil_crop3, cvRect(XX_awal,YY_awal,XX_akhir,YY_akhir));
    IplImage *tmp = cvCreateImage(cvGetSize(hasil_crop3),hasil_crop3->depth,hasil_crop3-
>nChannels);
    cvCopy(hasil_crop3, tmp, NULL);
    cvResetImageROI(hasil_crop3);
    hasil_crop3 = cvCloneImage(tmp);
    //=== Isi image tampung 3 ===
    if(tampung_3 == 0)
    {
        tampung_3 = cvCreateImage(cvSize(hasil_crop3->width, hasil_crop3->height), hasil_crop3-
>depth, 1);
    }
    if(tampung_3 != 0)
    {
        cvReleaseImage( &tampung_3 ); // fungsinya menghapus image jika tidak digunakan
        tampung_3 = cvCreateImage(cvSize(hasil_crop3->width, hasil_crop3->height), hasil_crop3-
>depth, 1);
    }
    tampung_3=cvCloneImage(hasil_crop3);
    //=====
    };
    //=====
    //=== Proses Crop image 4
    if (hasil_putar4 != 0)
    {
        if(hasil_crop4 == 0)
        {
            hasil_crop4 = cvCreateImage(cvSize(hasil_putar4->width,hasil_putar4->height),
hasil_putar4->depth, 1);
        }
        if(hasil_crop4 != 0)
        {
            cvReleaseImage( &hasil_crop4 ); // fungsinya menghapus image jika tidak digunakan
            hasil_crop4 = cvCreateImage(cvSize(hasil_putar4->width, hasil_putar4->height),
hasil_putar4->depth, 1);
        }
        hasil_crop4=cvCloneImage(hasil_putar4);
        //===== data koordinat area crop image 4
        int YY_awal = 25;
        int YY_akhir = 450;
        int XX_awal = 90;
        int XX_akhir = 555;
        //=====
        cvSetImageROI(hasil_crop4, cvRect(XX_awal,YY_awal,XX_akhir,YY_akhir));
        IplImage *tmp = cvCreateImage(cvGetSize(hasil_crop4),hasil_crop4->depth,hasil_crop4-
>nChannels);
        cvCopy(hasil_crop4, tmp, NULL);
        cvResetImageROI(hasil_crop4);
        hasil_crop4 = cvCloneImage(tmp);
        //=== Isi image tampung 4 ===
        if(tampung_4 == 0)
        {
            tampung_4 = cvCreateImage(cvSize(hasil_crop4->width, hasil_crop4->height), hasil_crop4-
>depth, 1);
        }
        if(tampung_4 != 0)
        {
            cvReleaseImage( &tampung_4 ); // fungsinya menghapus image jika tidak digunakan
            tampung_4 = cvCreateImage(cvSize(hasil_crop4->width, hasil_crop4->height), hasil_crop4-
>depth, 1);
        }
    }
}

```

```

tampung_4=cvCloneImage(hasil_crop4);
//=====
};
//=====
//== Proses Crop image 5
if (hasil_putar5 != 0)
{
    if(hasil_crop5 == 0)
    {
        hasil_crop5 = cvCreateImage(cvSize(hasil_putar5->width,hasil_putar5->height),
hasil_putar5->depth, 1);
    }
    if(hasil_crop5 != 0)
    {
        cvReleaseImage( &hasil_crop5 ); // fungsinya menghapus image jika tidak digunakan
        hasil_crop5 = cvCreateImage(cvSize(hasil_putar5->width, hasil_putar5->height),
hasil_putar5->depth, 1);
    }
    hasil_crop5=cvCloneImage(hasil_putar5);
    //===== data koordinat area crop image 5
    int YY_awal = 15;
    int YY_akhir = 445;
    int XX_awal = 50;
    int XX_akhir = 508;
    //=====
    cvSetImageROI(hasil_crop5, cvRect(XX_awal,YY_awal,XX_akhir,YY_akhir));
    IplImage *tmp = cvCreateImage(cvGetSize(hasil_crop5),hasil_crop5->depth,hasil_crop5-
>nChannels);
    cvCopy(hasil_crop5, tmp, NULL);
    cvResetImageROI(hasil_crop5);
    hasil_crop5 = cvCloneImage(tmp);
    //== Isi image tampung 5 ==
    if(tampung_5 == 0)
    {
        tampung_5 = cvCreateImage(cvSize(hasil_crop5->width, hasil_crop5->height), hasil_crop5-
>depth, 1);
    }
    if(tampung_5 != 0)
    {
        cvReleaseImage( &tampung_5 ); // fungsinya menghapus image jika tidak digunakan
        tampung_5 = cvCreateImage(cvSize(hasil_crop5->width, hasil_crop5->height), hasil_crop5-
>depth, 1);
    }
    tampung_5=cvCloneImage(hasil_crop5);
    //=====
};
//=====
//== Proses Crop image 6
if (hasil_putar6 != 0)
{
    if(hasil_crop6 == 0)
    {
        hasil_crop6 = cvCreateImage(cvSize(hasil_putar6->width,hasil_putar6->height),
hasil_putar6->depth, 1);
    }
    if(hasil_crop6 != 0)
    {
        cvReleaseImage( &hasil_crop6 ); // fungsinya menghapus image jika tidak digunakan
        hasil_crop6 = cvCreateImage(cvSize(hasil_putar6->width, hasil_putar6->height),
hasil_putar6->depth, 1);
    }
    hasil_crop6=cvCloneImage(hasil_putar6);
    //===== data koordinat area crop image 6
    int YY_awal = 40;
    int YY_akhir = 400;
    int XX_awal = 92;
    int XX_akhir = 560;
    //=====
    cvSetImageROI(hasil_crop6, cvRect(XX_awal,YY_awal,XX_akhir,YY_akhir));
    IplImage *tmp = cvCreateImage(cvGetSize(hasil_crop6),hasil_crop6->depth,hasil_crop6-
>nChannels);

```

```

cvCopy(hasil_crop6, tmp, NULL);
cvResetImageROI(hasil_crop6);
hasil_crop6 = cvCloneImage(tmp);
//=== Isi image tampung 6 ===
if(tampung_6 == 0)
{
    tampung_6 = cvCreateImage(cvSize(hasil_crop6->width, hasil_crop6->height), hasil_crop6->depth, 1);
}
if(tampung_6 != 0)
{
    cvReleaseImage( &tampung_6 ); // fungsinya menghapus image jika tidak digunakan
    tampung_6 = cvCreateImage(cvSize(hasil_crop6->width, hasil_crop6->height), hasil_crop6->depth, 1);
}
tampung_6=cvCloneImage(hasil_crop6);
//=====
};
};
//=====
        if (pilih_operasi==0){this->button_tampilkan_RGB_Click_1(sender, e);}
        if (pilih_operasi==1){this->btn_gray_auto_Click(sender, e);}
}

// button proses grayscale otomatis
private: System::Void btn_gray_auto_Click(System::Object^ sender, System::EventArgs^ e)
{
    // Proses dari kamera
    if (status_sumber==1)
    {
        // === Proses grayscale image 1 =====
        if (hasil_crop1 != 0)
        {
            if(grayimage1 == 0)
            {
                grayimage1 = cvCreateImage(cvSize(hasil_crop1->width, hasil_crop1->height),
                hasil_crop1->depth, 1);
            }
            if(grayimage1 != 0)
            {
                cvReleaseImage( &grayimage1 ); // fungsinya menghapus image jika tidak digunakan
                grayimage1 = cvCreateImage(cvSize(hasil_crop1->width, hasil_crop1->height),
                hasil_crop1->depth, 1);
            }
            //convert to grayscale
            cvCvtColor(hasil_crop1, grayimage1, CV_RGB2GRAY);
            //=== Isi image tampung 1 ===
            if(tampung_1 == 0)
            {
                tampung_1 = cvCreateImage(cvSize(grayimage1->width, grayimage1->height),
                grayimage1->depth, 1);
            }
            if(tampung_1 != 0)
            {
                cvReleaseImage( &tampung_1 ); // fungsinya menghapus image jika tidak digunakan
                tampung_1 = cvCreateImage(cvSize(grayimage1->width, grayimage1->height),
                grayimage1->depth, 1);
            }
            tampung_1=cvCloneImage(grayimage1);
        }
        //=====
        // === Proses grayscale image 2 =====
        if (hasil_crop2 != 0)
        {
            if(grayimage2 == 0)
            {
                grayimage2 = cvCreateImage(cvSize(hasil_crop2->width, hasil_crop2->height),
                hasil_crop2->depth, 1);
            }
            if(grayimage2 != 0)
            {

```

```

        cvReleaseImage( &grayimage2 ); // fungsinya menghapus image jika tidak digunakan
        grayimage2 = cvCreateImage(cvSize(hasil_crop2->width, hasil_crop2->height),
hasil_crop2->depth, 1);
    }
    //convert to grayscale
    cvCvtColor(hasil_crop2, grayimage2, CV_RGB2GRAY);
    //=== Isi image tampung 2 ===
    if(tampung_2 == 0)
    {
        tampung_2 = cvCreateImage(cvSize(grayimage2->width, grayimage2->height),
grayimage2->depth, 1);
    }
    if(tampung_2 != 0)
    {
        cvReleaseImage( &tampung_2 ); // fungsinya menghapus image jika tidak digunakan
        tampung_2 = cvCreateImage(cvSize(grayimage2->width, grayimage2->height),
grayimage2->depth, 1);
    }
    tampung_2=cvCloneImage(grayimage2);
    }
    //=====
    // ==== Proses grayscale image 3 =====
    if (hasil_crop3 != 0)
    {
        if(grayimage3 == 0)
        {
            grayimage3 = cvCreateImage(cvSize(hasil_crop3->width, hasil_crop3->height),
hasil_crop3->depth, 1);
        }
        if(grayimage3 != 0)
        {
            cvReleaseImage( &grayimage3 ); // fungsinya menghapus image jika tidak digunakan
            grayimage3 = cvCreateImage(cvSize(hasil_crop3->width, hasil_crop3->height),
hasil_crop3->depth, 1);
        }
        //convert to grayscale
        cvCvtColor(hasil_crop3, grayimage3, CV_RGB2GRAY);
        //=== Isi image tampung 3 ===
        if(tampung_3 == 0)
        {
            tampung_3 = cvCreateImage(cvSize(grayimage3->width, grayimage3->height),
grayimage3->depth, 1);
        }
        if(tampung_3 != 0)
        {
            cvReleaseImage( &tampung_3 ); // fungsinya menghapus image jika tidak digunakan
            tampung_3 = cvCreateImage(cvSize(grayimage3->width, grayimage3->height),
grayimage3->depth, 1);
        }
        tampung_3=cvCloneImage(grayimage3);
    }
    //=====
    // ==== Proses grayscale image 4 =====
    if (hasil_crop4 != 0)
    {
        if(grayimage4 == 0)
        {
            grayimage4 = cvCreateImage(cvSize(hasil_crop4->width, hasil_crop4->height),
hasil_crop4->depth, 1);
        }
        if(grayimage4 != 0)
        {
            cvReleaseImage( &grayimage4 ); // fungsinya menghapus image jika tidak digunakan
            grayimage4 = cvCreateImage(cvSize(hasil_crop4->width, hasil_crop4->height),
hasil_crop4->depth, 1);
        }
        //convert to grayscale
        cvCvtColor(hasil_crop4, grayimage4, CV_RGB2GRAY);
        //=== Isi image tampung 4 ===
        if(tampung_4 == 0)
        {

```

```

        tampung_4 = cvCreateImage(cvSize(grayimage4->width, grayimage4->height),
grayimage4->depth, 1);
    }
    if(tampung_4 != 0)
    {
        cvReleaseImage( &tampung_4 ); // fungsinya menghapus image jika tidak digunakan
        tampung_4 = cvCreateImage(cvSize(grayimage4->width, grayimage4->height),
grayimage4->depth, 1);
    }
    tampung_4=cvCloneImage(grayimage4);
    }
    //=====
    // ==== Proses grayscale image 5 =====
    if (hasil_crop5 != 0)
    {
        if(grayimage5 == 0)
        {
            grayimage5 = cvCreateImage(cvSize(hasil_crop5->width, hasil_crop5->height),
hasil_crop5->depth, 1);
        }
        if(grayimage5 != 0)
        {
            cvReleaseImage( &grayimage5 ); // fungsinya menghapus image jika tidak digunakan
            grayimage5 = cvCreateImage(cvSize(hasil_crop5->width, hasil_crop5->height),
hasil_crop5->depth, 1);
        }
        //convert to grayscale
        cvCvtColor(hasil_crop5, grayimage5, CV_RGB2GRAY);
        //=== Isi image tampung 5 ===
        if(tampung_5 == 0)
        {
            tampung_5 = cvCreateImage(cvSize(grayimage5->width, grayimage5->height),
grayimage5->depth, 1);
        }
        if(tampung_5 != 0)
        {
            cvReleaseImage( &tampung_5 ); // fungsinya menghapus image jika tidak digunakan
            tampung_5 = cvCreateImage(cvSize(grayimage5->width, grayimage5->height),
grayimage5->depth, 1);
        }
        tampung_5=cvCloneImage(grayimage5);
    }
    //=====
    // ==== Proses grayscale image 6 =====
    if (hasil_crop6 != 0)
    {
        if(grayimage6 == 0)
        {
            grayimage6 = cvCreateImage(cvSize(hasil_crop6->width, hasil_crop6->height),
hasil_crop6->depth, 1);
        }
        if(grayimage6 != 0)
        {
            cvReleaseImage( &grayimage6 ); // fungsinya menghapus image jika tidak digunakan
            grayimage6 = cvCreateImage(cvSize(hasil_crop6->width, hasil_crop6->height),
hasil_crop6->depth, 1);
        }
        //convert to grayscale
        cvCvtColor(hasil_crop6, grayimage6, CV_RGB2GRAY);
        //=== Isi image tampung 6 ===
        if(tampung_6 == 0)
        {
            tampung_6 = cvCreateImage(cvSize(grayimage6->width, grayimage6->height),
grayimage6->depth, 1);
        }
        if(tampung_6 != 0)
        {
            cvReleaseImage( &tampung_6 ); // fungsinya menghapus image jika tidak digunakan
            tampung_6 = cvCreateImage(cvSize(grayimage6->width, grayimage6->height),
grayimage6->depth, 1);
        }
    }
}

```

```

tampung_6=cvCloneImage(grayimage6);
}
//=====
if (pilih_operasi==0){this->button_tampilkan_Click(sender, e);}
if (pilih_operasi==1){this->btn_thresholding_auto_Click(sender, e);}
} // akhir proses dari kamera
//=====
// Proses dari gambar scan
if (status_sumber==0)
{
    // === Proses grayscale ===
    if (origimage != 0)
    {
        if(grayimage == 0)
        {
            grayimage = cvCreateImage(cvSize(origimage->width, origimage->height), origimage-
>depth, 1);
        }
        if(grayimage != 0)
        {
            cvReleaseImage( &grayimage ); // fungsinya menghapus image jika tidak digunakan
            grayimage = cvCreateImage(cvSize(origimage->width, origimage->height), origimage-
>depth, 1);
        }
        //convert to grayscale
        cvCvtColor(origimage, grayimage, CV_RGB2GRAY);
        //=== Tampilkan image ===
        if(tampil_1 == 0)
        {
            tampil_1 = cvCreateImage( cvGetSize(grayimage), 8, 3 );
        }
        if(tampil_1 != 0)
        {
            cvReleaseImage( &tampil_1 ); // fungsinya menghapus image jika tidak digunakan
            tampil_1 = cvCreateImage( cvGetSize(grayimage), 8, 3 );
        }
        //=====
        cvCvtColor(grayimage, tampil_1, CV_GRAY2BGR);
        pictureBox_1->Image = gcnew
        System::Drawing::Bitmap(tampil_1->width,tampil_1->height,tampil_1->widthStep,
        System::Drawing::Imaging::PixelFormat::Format24bppRgb,(System::IntPtr)tampil_1-
>imageData);
        pictureBox_1->Refresh();
        if (pilih_operasi==1){this->btn_thresholding_auto_Click(sender, e);}
    }
}

// Proses Thresholding dengan Adaptive Threshold
private: System::Void btn_thresholding_auto_Click(System::Object^ sender,
System::EventArgs^ e) {
    // Proses dari kamera
    if (status_sumber==1)
    {
        //===== Proses Thresholding Image 1 =====
        if (grayimage1 !=0)
        {
            if(binaryimage1 == 0)
            {
                binaryimage1 = cvCreateImage(cvSize(grayimage1->width, grayimage1->height),
grayimage1->depth, 1);
            }
            if(binaryimage1 != 0)
            {
                cvReleaseImage( &binaryimage1 ); // fungsinya menghapus image jika tidak digunakan
                binaryimage1 = cvCreateImage(cvSize(grayimage1->width, grayimage1->height),
grayimage1->depth, 1);
            }
            int nilai1 = 57;
            int nilai2 = 3;
            // === PROSES THRESHOLD DENGAN ADAPTIVE THRESHOLD

```



```

        cvAdaptiveThreshold(grayimage1, binaryimage1, 255, CV_ADAPTIVE_THRESH_GAUSSIAN_C,
CV_THRESH_BINARY, nilai1, nilai2);
        //=== Isi image tampung 1 ===
        if(tampung_1 == 0)
        {
            tampung_1 = cvCreateImage(cvSize(binaryimage1->width, binaryimage1-
>height), binaryimage1->depth, 1);
        }
        if(tampung_1 != 0)
        {
            cvReleaseImage( &tampung_1 ); // fungsinya menghapus image jika tidak
digunakan
            tampung_1 = cvCreateImage(cvSize(binaryimage1->width, binaryimage1-
>height), binaryimage1->depth, 1);
        }
        tampung_1=cvCloneImage(binaryimage1);
        //=====
    };

    //=====
    //===== Proses Thresholding Image 2 =====
    if (grayimage2 !=0)
    {
        if(binaryimage2 == 0)
        {
            binaryimage2 = cvCreateImage(cvSize(grayimage2->width, grayimage2->height),
grayimage2->depth, 1);
        }
        if(binaryimage2 != 0)
        {
            cvReleaseImage( &binaryimage2 ); // fungsinya menghapus image jika tidak digunakan
            binaryimage2 = cvCreateImage(cvSize(grayimage2->width, grayimage2->height),
grayimage2->depth, 1);
        }
        int nilai1 = 57;
        int nilai2 = 3;
        // === PROSES THRESHOLD DENGAN ADAPTIVE THRESHOLD
        cvAdaptiveThreshold(grayimage2, binaryimage2, 255, CV_ADAPTIVE_THRESH_GAUSSIAN_C,
CV_THRESH_BINARY, nilai1, nilai2);
        //=== Isi image tampung 2 ===
        if(tampung_2 == 0)
        {
            tampung_2 = cvCreateImage(cvSize(binaryimage2->width, binaryimage2-
>height), binaryimage2->depth, 1);
        }
        if(tampung_2 != 0)
        {
            cvReleaseImage( &tampung_2 ); // fungsinya menghapus image jika tidak
digunakan
            tampung_2 = cvCreateImage(cvSize(binaryimage2->width, binaryimage2-
>height), binaryimage2->depth, 1);
        }
        tampung_2=cvCloneImage(binaryimage2);
        //=====
    };

    //=====
    //===== Proses Thresholding Image 3 =====
    if (grayimage3 !=0)
    {
        if(binaryimage3 == 0)
        {
            binaryimage3 = cvCreateImage(cvSize(grayimage3->width, grayimage3->height),
grayimage3->depth, 1);
        }
        if(binaryimage3 != 0)
        {
            cvReleaseImage( &binaryimage3 ); // fungsinya menghapus image jika tidak digunakan
            binaryimage3 = cvCreateImage(cvSize(grayimage3->width, grayimage3->height),
grayimage3->depth, 1);
        }
        int nilai1 = 57;

```

```

        int nilai2 = 3;
        // === PROSES THRESHOLD DENGAN ADAPTIVE THRESHOLD
        cvAdaptiveThreshold(grayimage3, binaryimage3, 255, CV_ADAPTIVE_THRESH_GAUSSIAN_C,
CV_THRESH_BINARY, nilai1, nilai2);
        //=== Isi image tampung 3 ===
        if(tampung_3 == 0)
        {
            tampung_3 = cvCreateImage(cvSize(binaryimage3->width, binaryimage3-
>height), binaryimage3->depth, 1);
        }
        if(tampung_3 != 0)
        {
            cvReleaseImage( &tampung_3 ); // fungsinya menghapus image jika tidak
digunakan
            tampung_3 = cvCreateImage(cvSize(binaryimage3->width, binaryimage3-
>height), binaryimage3->depth, 1);
        }
        tampung_3=cvCloneImage(binaryimage3);
        //=====
    };
    //=====
    //===== Proses Thresholding Image 4 =====
    if (grayimage4 !=0)
    {
        if(binaryimage4 == 0)
        {
            binaryimage4 = cvCreateImage(cvSize(grayimage4->width, grayimage4->height),
grayimage4->depth, 1);
        }
        if(binaryimage4 != 0)
        {
            cvReleaseImage( &binaryimage4 ); // fungsinya menghapus image jika tidak digunakan
            binaryimage4 = cvCreateImage(cvSize(grayimage4->width, grayimage4->height),
grayimage4->depth, 1);
        }
        int nilai1 = 57;
        int nilai2 = 3;
        // === PROSES THRESHOLD DENGAN ADAPTIVE THRESHOLD
        cvAdaptiveThreshold(grayimage4, binaryimage4, 255, CV_ADAPTIVE_THRESH_GAUSSIAN_C,
CV_THRESH_BINARY, nilai1, nilai2);
        //=== Isi image tampung 4 ===
        if(tampung_4 == 0)
        {
            tampung_4 = cvCreateImage(cvSize(binaryimage4->width, binaryimage4-
>height), binaryimage4->depth, 1);
        }
        if(tampung_4 != 0)
        {
            cvReleaseImage( &tampung_4 ); // fungsinya menghapus image jika tidak
digunakan
            tampung_4 = cvCreateImage(cvSize(binaryimage4->width, binaryimage4-
>height), binaryimage4->depth, 1);
        }
        tampung_4=cvCloneImage(binaryimage4);
        //=====
    };
    //=====
    //===== Proses Thresholding Image 5 =====
    if (grayimage5 !=0)
    {
        if(binaryimage5 == 0)
        {
            binaryimage5 = cvCreateImage(cvSize(grayimage5->width, grayimage5->height),
grayimage5->depth, 1);
        }
        if(binaryimage5 != 0)
        {
            cvReleaseImage( &binaryimage5 ); // fungsinya menghapus image jika tidak digunakan
            binaryimage5 = cvCreateImage(cvSize(grayimage5->width, grayimage5->height),
grayimage5->depth, 1);
        }
    }

```

```

        int nilai1 = 57;
        int nilai2 = 3;
        // === PROSES THRESHOLD DENGAN ADAPTIVE THRESHOLD
        cvAdaptiveThreshold(grayimage5, binaryimage5, 255, CV_ADAPTIVE_THRESH_GAUSSIAN_C,
CV_THRESH_BINARY, nilai1, nilai2);
        //=== Isi image tampung 5 ===
        if(tampung_5 == 0)
        {
            tampung_5 = cvCreateImage(cvSize(binaryimage5->width, binaryimage5-
>height), binaryimage5->depth, 1);
        }
        if(tampung_5 != 0)
        {
            cvReleaseImage( &tampung_5 ); // fungsinya menghapus image jika tidak
digunakan
            tampung_5 = cvCreateImage(cvSize(binaryimage5->width, binaryimage5-
>height), binaryimage5->depth, 1);
        }
        tampung_5=cvCloneImage(binaryimage5);
        //=====
    };
    //=====
    //===== Proses Thresholding Image 6 =====
    if (grayimage6 !=0)
    {
        if(binaryimage6 == 0)
        {
            binaryimage6 = cvCreateImage(cvSize(grayimage6->width, grayimage6->height),
grayimage6->depth, 1);
        }
        if(binaryimage6 != 0)
        {
            cvReleaseImage( &binaryimage6 ); // fungsinya menghapus image jika tidak digunakan
            binaryimage6 = cvCreateImage(cvSize(grayimage6->width, grayimage6->height),
grayimage6->depth, 1);
        }
        int nilai1 = 57;
        int nilai2 = 3;
        // === PROSES THRESHOLD DENGAN ADAPTIVE THRESHOLD
        cvAdaptiveThreshold(grayimage6, binaryimage6, 255, CV_ADAPTIVE_THRESH_GAUSSIAN_C,
CV_THRESH_BINARY, nilai1, nilai2);
        //=== Isi image tampung 6 ===
        if(tampung_6 == 0)
        {
            tampung_6 = cvCreateImage(cvSize(binaryimage6->width, binaryimage6-
>height), binaryimage6->depth, 1);
        }
        if(tampung_6 != 0)
        {
            cvReleaseImage( &tampung_6 ); // fungsinya menghapus image jika tidak
digunakan
            tampung_6 = cvCreateImage(cvSize(binaryimage6->width, binaryimage6-
>height), binaryimage6->depth, 1);
        }
        tampung_6=cvCloneImage(binaryimage6);
        //=====
    };
    //=====
    if (pilih_operasi==0){this->button_tampilkan_Click(sender, e);}
    if (pilih_operasi==1){this->btn_dilation_erotion_Click(sender, e);}
} // akhir proses dari kamera
//=====
// Proses gambar dari scan
if (status_sumber==0)
{
    //===== Proses Thresholding =====
    if (grayimage !=0)
    {
        if(binaryimage == 0)
        {

```

```

        binaryimage = cvCreateImage(cvSize(grayimage->width, grayimage->height),
grayimage->depth, 1);
    }
    if(binaryimage != 0)
    {
        cvReleaseImage( &binaryimage ); // fungsinya menghapus image jika tidak digunakan
        binaryimage = cvCreateImage(cvSize(grayimage->width, grayimage->height),
grayimage->depth, 1);
    }
    //=====
    // cetakan 1 sisi
    if (jenis_cetakan==1)
    {
        cvThreshold(grayimage, binaryimage, 238, 255, CV_THRESH_BINARY);
    }
    // cetakan 2 sisi
    if (jenis_cetakan==2)
    {
        cvThreshold(grayimage, binaryimage, 190, 255, CV_THRESH_BINARY);
    }
    //== Tampilkan image ==
    if(tampil_1 == 0)
    {
        tampil_1 = cvCreateImage( cvGetSize(binaryimage), 8, 3 );
    }
    if(tampil_1 != 0)
    {
        cvReleaseImage( &tampil_1 ); // fungsinya menghapus image jika tidak digunakan
        tampil_1 = cvCreateImage( cvGetSize(binaryimage), 8, 3 );
    }
    //=====
    cvCvtColor(binaryimage, tampil_1, CV_GRAY2BGR);

pictureBox_1->Image = gcnew
System::Drawing::Bitmap(tampil_1->width,tampil_1->height,tampil_1->widthStep,
System::Drawing::Imaging::PixelFormat::Format24bppRgb,(System::IntPtr)tampil_1-
>imageData);
pictureBox_1->Refresh();
if (pilih_operasi==1){this->btn_dilation_erotion_Click(sender, e);}
    }
    } // akhir proses dari scan
}

// buttton tampilkan
private: System::Void button_tampilkan_Click(System::Object^ sender, System::EventArgs^
e) {

    //===== Tampilkan image 1 =====
    if (tampung_1 !=0)
    {
        if(tampil_1 == 0)
        {
            tampil_1 = cvCreateImage( cvGetSize(tampung_1), 8, 3 );
        }
        if(tampil_1 != 0)
        {
            cvReleaseImage( &tampil_1 ); // fungsinya menghapus image jika tidak digunakan
            tampil_1 = cvCreateImage( cvGetSize(tampung_1), 8, 3 );
        }
    }
    //=====
    cvCvtColor(tampung_1, tampil_1, CV_GRAY2BGR);
pictureBox_1->Image = gcnew
System::Drawing::Bitmap(tampil_1->width,tampil_1->height,tampil_1->widthStep,
System::Drawing::Imaging::PixelFormat::Format24bppRgb,(System::IntPtr)tampil_1-
>imageData);
pictureBox_1->Refresh();
}
//=====
//===== Tampilkan image 2 =====
if (tampung_2 !=0)
{

```

```

        if(tampil_2 == 0)
        {
            tampil_2 = cvCreateImage( cvGetSize(tampung_2), 8, 3 );
        }
        if(tampil_2 != 0)
        {
            cvReleaseImage( &tampil_2 ); // fungsinya menghapus image jika tidak digunakan
            tampil_2 = cvCreateImage( cvGetSize(tampung_2), 8, 3 );
        }
//=====
        cvCvtColor(tampung_2, tampil_2, CV_GRAY2BGR);
pictureBox_2->Image = gcnew
System::Drawing::Bitmap(tampil_2->width,tampil_2->height,tampil_2->widthStep,
System::Drawing::Imaging::PixelFormat::Format24bppRgb,(System::IntPtr)tampil_2-
>imageData);
pictureBox_2->Refresh();
}
//=====
//===== Tampilkan image 3 =====
if (tampung_3 !=0)
{
    if(tampil_3 == 0)
    {
        tampil_3 = cvCreateImage( cvGetSize(tampung_3), 8, 3 );
    }
    if(tampil_3 != 0)
    {
        cvReleaseImage( &tampil_3 ); // fungsinya menghapus image jika tidak digunakan
        tampil_3 = cvCreateImage( cvGetSize(tampung_3), 8, 3 );
    }
//=====
        cvCvtColor(tampung_3, tampil_3, CV_GRAY2BGR);
pictureBox_3->Image = gcnew
System::Drawing::Bitmap(tampil_3->width,tampil_3->height,tampil_3->widthStep,
System::Drawing::Imaging::PixelFormat::Format24bppRgb,(System::IntPtr)tampil_3-
>imageData);
pictureBox_3->Refresh();
}
//=====
//===== Tampilkan image 4 =====
if (tampung_4 !=0)
{
    if(tampil_4 == 0)
    {
        tampil_4 = cvCreateImage( cvGetSize(tampung_4), 8, 3 );
    }
    if(tampil_4 != 0)
    {
        cvReleaseImage( &tampil_4 ); // fungsinya menghapus image jika tidak digunakan
        tampil_4 = cvCreateImage( cvGetSize(tampung_4), 8, 3 );
    }
//=====
        cvCvtColor(tampung_4, tampil_4, CV_GRAY2BGR);
pictureBox_4->Image = gcnew
System::Drawing::Bitmap(tampil_4->width,tampil_4->height,tampil_4->widthStep,
System::Drawing::Imaging::PixelFormat::Format24bppRgb,(System::IntPtr)tampil_4-
>imageData);
pictureBox_4->Refresh();
}
//=====
//===== Tampilkan image 5 =====
if (tampung_5 !=0)
{
    if(tampil_5 == 0)
    {
        tampil_5 = cvCreateImage( cvGetSize(tampung_5), 8, 3 );
    }
    if(tampil_5 != 0)
    {
        cvReleaseImage( &tampil_5 ); // fungsinya menghapus image jika tidak digunakan
        tampil_5 = cvCreateImage( cvGetSize(tampung_5), 8, 3 );
    }
}

```

```

    }
    //=====
    cvCvtColor(tampung_5, tampil_5, CV_GRAY2BGR);
    pictureBox_5->Image = gcnnew
    System::Drawing::Bitmap(tampil_5->width,tampil_5->height,tampil_5->widthStep,
    System::Drawing::Imaging::PixelFormat::Format24bppRgb,(System::IntPtr)tampil_5-
    >imageData);
    pictureBox_5->Refresh();
    }
    //=====
    //===== Tampilkan image 6 =====
    if (tampung_6 !=0)
    {
        if(tampil_6 == 0)
        {
            tampil_6 = cvCreateImage( cvGetSize(tampung_6), 8, 3 );
        }
        if(tampil_6 != 0)
        {
            cvReleaseImage( &tampil_6 ); // fungsinya menghapus image jika tidak digunakan
            tampil_6 = cvCreateImage( cvGetSize(tampung_6), 8, 3 );
        }
    }
    //=====
    cvCvtColor(tampung_6, tampil_6, CV_GRAY2BGR);
    pictureBox_6->Image = gcnnew
    System::Drawing::Bitmap(tampil_6->width,tampil_6->height,tampil_6->widthStep,
    System::Drawing::Imaging::PixelFormat::Format24bppRgb,(System::IntPtr)tampil_6-
    >imageData);
    pictureBox_6->Refresh();
    }
    // Button Proses Dilation dan Erotion
    private: System::Void btn_dilation_erotion_Click(System::Object^ sender,
    System::EventArgs^ e) {
        // Proses dari kamera
        if (status_sumber==1)
        {
            int banyak_dilation=1;
            int banyak_erotion=2;
            // ===== proses dilation dan erotion image 1
            cvDilate(binaryimage1, binaryimage1,0,banyak_dilation); //proses Dilate
            cvErode(binaryimage1, binaryimage1,0,banyak_erotion); //proses Erode
            //=== Isi image tampung 1 ===
            if(tampung_1 == 0)
            {
                tampung_1 = cvCreateImage(cvSize(binaryimage1->width, binaryimage1->height),
                binaryimage1->depth, 1);
            }
            if(tampung_1 != 0)
            {
                cvReleaseImage( &tampung_1 ); // fungsinya menghapus image jika tidak digunakan
                tampung_1 = cvCreateImage(cvSize(binaryimage1->width, binaryimage1->height),
                binaryimage1->depth, 1);
            }
            tampung_1=cvCloneImage(binaryimage1);
            //=====
            // ===== proses dilation dan erotion image 2
            cvDilate(binaryimage2, binaryimage2,0,banyak_dilation); //proses Dilate
            cvErode(binaryimage2, binaryimage2,0,banyak_erotion); //proses Erode
            //=== Isi image tampung 2 ===
            if(tampung_2 == 0)
            {
                tampung_2 = cvCreateImage(cvSize(binaryimage2->width, binaryimage2->height),
                binaryimage2->depth, 1);
            }
            if(tampung_2 != 0)
            {
                cvReleaseImage( &tampung_2 ); // fungsinya menghapus image jika tidak digunakan
                tampung_2 = cvCreateImage(cvSize(binaryimage2->width, binaryimage2->height),
                binaryimage2->depth, 1);
            }
        }
    }

```

```

tampung_2=cvCloneImage(binaryimage2);
//=====
// ===== proses dilation dan erotion image 3
cvDilate(binaryimage3, binaryimage3,0,banyak_dilation);//proses Dilate
cvErode(binaryimage3, binaryimage3,0,banyak_erotion);//proses Erode
//=== Isi image tampung 3 ===
if(tampung_3 == 0)
{
    tampung_3 = cvCreateImage(cvSize(binaryimage3->width, binaryimage3->height),
binaryimage3->depth, 1);
}
if(tampung_3 != 0)
{
    cvReleaseImage( &tampung_3 ); // fungsinya menghapus image jika tidak digunakan
    tampung_3 = cvCreateImage(cvSize(binaryimage3->width, binaryimage3->height),
binaryimage3->depth, 1);
}
    tampung_3=cvCloneImage(binaryimage3);
//=====
// ===== proses dilation dan erotion image 4
cvDilate(binaryimage4, binaryimage4,0,banyak_dilation);//proses Dilate
cvErode(binaryimage4, binaryimage4,0,banyak_erotion);//proses Erode
//=== Isi image tampung 4 ===
if(tampung_4 == 0)
{
    tampung_4 = cvCreateImage(cvSize(binaryimage4->width, binaryimage4->height),
binaryimage4->depth, 1);
}
if(tampung_4 != 0)
{
    cvReleaseImage( &tampung_4 ); // fungsinya menghapus image jika tidak digunakan
    tampung_4 = cvCreateImage(cvSize(binaryimage4->width, binaryimage4->height),
binaryimage4->depth, 1);
}
    tampung_4=cvCloneImage(binaryimage4);
//=====
// ===== proses dilation dan erotion image 5
cvDilate(binaryimage5, binaryimage5,0,banyak_dilation);//proses Dilate
cvErode(binaryimage5, binaryimage5,0,banyak_erotion);//proses Erode
//=== Isi image tampung 5 ===
if(tampung_5 == 0)
{
    tampung_5 = cvCreateImage(cvSize(binaryimage5->width, binaryimage5->height),
binaryimage5->depth, 1);
}
if(tampung_5 != 0)
{
    cvReleaseImage( &tampung_5 ); // fungsinya menghapus image jika tidak digunakan
    tampung_5 = cvCreateImage(cvSize(binaryimage5->width, binaryimage5->height),
binaryimage5->depth, 1);
}
    tampung_5=cvCloneImage(binaryimage5);
//=====
// ===== proses dilation dan erotion image 6
cvDilate(binaryimage6, binaryimage6,0,banyak_dilation);//proses Dilate
cvErode(binaryimage6, binaryimage6,0,banyak_erotion);//proses Erode
//=== Isi image tampung 6 ===
if(tampung_6 == 0)
{
    tampung_6 = cvCreateImage(cvSize(binaryimage6->width, binaryimage6->height),
binaryimage6->depth, 1);
}
if(tampung_6 != 0)
{
    cvReleaseImage( &tampung_6 ); // fungsinya menghapus image jika tidak digunakan
    tampung_6 = cvCreateImage(cvSize(binaryimage6->width, binaryimage6->height),
binaryimage6->depth, 1);
}
    tampung_6=cvCloneImage(binaryimage6);
//=====
if (pilih_operasi==0){this->button_tampilkan_Click(sender, e);}

```

```

if (pilih_operasi==1){this->btn_baca_titik_Click(sender, e);}
} // akhir proses kamera
//=====
// proses dari gambar scan
if (status_sumber==0)
{
    if(erot == 0)
    {
        erot = cvCreateImage(cvSize(binaryimage->width, binaryimage->height),
binaryimage->depth, 1);
    };
    if(erot != 0)
    {
        cvReleaseImage( &erot ); // fungsinya menghapus image jika tidak digunakan
        erot = cvCreateImage(cvSize(binaryimage->width,
binaryimage->height), binaryimage->depth, 1);
    };
    //=====
    cvErode(binaryimage, erot,0,1); //proses Erode perbesaran titik 1x
    //=== Tampilkan image ===
    if(tampil_1 == 0)
    {
        tampil_1 = cvCreateImage( cvGetSize(erot), 8, 3 );
    }
    if(tampil_1 != 0)
    {
        cvReleaseImage( &tampil_1 ); // fungsinya menghapus image jika tidak digunakan
        tampil_1 = cvCreateImage( cvGetSize(erot), 8, 3 );
    }
    cvCvtColor(erot, tampil_1, CV_GRAY2BGR);
    pictureBox_1->Image = gcnew
    System::Drawing::Bitmap(tampil_1->width,tampil_1->height,tampil_1->widthStep,
    System::Drawing::Imaging::PixelFormat::Format24bppRgb,(System::IntPtr)tampil_1-
>imageData);
    pictureBox_1->Refresh();
    if (pilih_operasi==1){this->btn_baca_titik_Click(sender, e);}
} // akhir proses dari scan
}

// button baca titik dot HITAM (PROSES BLOB ANALYSIS)
private: System::Void btn_baca_titik_Click(System::Object^ sender, System::EventArgs^ e)
{
    // proses dari kamera
    if (status_sumber==1)
    {
        int count_lama=0;
        // BLOB ANALYSIS Image 1
        IplImage **dst_img;
        CvMemStorage *storage = cvCreateMemStorage (0);
        CvSeq *contours = 0;
        cvClearMemStorage( storage );
        int levels = 0;
        dst_img = (IplImage **) cvAlloc (sizeof (IplImage *) * 3);
        for (i = 0; i < 3; i++) {
            dst_img[i] = cvCloneImage (binaryimage1);};
        int Nc = cvFindContours(binaryimage1,storage,&contours,sizeof (CvContour),CV_RETR_LIST);
        int batas = 5;
        int tot=0;
        //=====
        StreamWriter^ outFile = gcnew StreamWriter ("Data_XY_image1.txt");
        //=====
        CvPoint mins[1000],maxs[1000];
        CvPoint minsy[1000],maxsy[1000];
        int n=0,pos=0,ulang=0; double buf=0;
        int mak1=1,mak2=0,n1=0,n2=0,i=0;
        //=====
        for (i=0;i<1000;i++){garis_x1[i]=0;garis_y1[i]=0;};
        count1=0;
        for( CvSeq* c=contours; c!=NULL; c=c->h_next )

```



```

{
    tot=c->total;
    if(tot>batas){count1++;
};
//=====
int total_y1[1000], total_x1[1000];
int maxx=0,maxys=0,minx=1000,minys=1000;
if(c->total>mak1){mak1=c->total;n1=n;}
for( i=0; i<c->total; i=i+1 )
{
    CvPoint* p = CV_GET_SEQ_ELEM( CvPoint, c, i );
    if(maxx<p->x)
    {
        maxs[n].x=p->x;
        maxs[n].y=p->y;
        maxx=p->x;
    }
    if(minx>p->x)
    {
        mins[n].x=p->x;
        mins[n].y=p->y;
        minx=p->x;
    }
    if(maxys<p->y)
    {
        maxsy[n].x=p->x;
        maxsy[n].y=p->y;
        maxys=p->y;}
    if(minys>p->y)
    {
        minsy[n].x=p->x;
        minsy[n].y=p->y;
        minys=p->y;
    }
}

if (count_lama != count1)
{
    count_lama=count1;
    rata_x1[count1]=(minx+maxx)/2;
    rata_y1[count1]=(minys+maxys)/2;
    //==
    outFile->Write("Dot ke: ");
    outFile->Write(Convert::ToString(count1));
    outFile->Write(" >>> X= ");
    outFile->Write(Convert::ToString(rata_x1[count1]));
    outFile->Write(", Y= ");
    outFile->Write(Convert::ToString(rata_y1[count1]));
    outFile->Write("\r\n");
}
n++;
}
outFile->Write("Banyak Dot: ");
outFile->Write(Convert::ToString(count1));
outFile->Write("\r\n");
outFile->Close();
//=====
cvDrawContours (dst_img[0], contours, CV_RGB (255, 0, 0), CV_RGB (0, 255, 0), levels - 1,
2, CV_AA, cvPoint (0, 0));
cvDrawContours (dst_img[1], contours, CV_RGB (255, 0, 0), CV_RGB (0, 255, 0), levels, 2,
CV_AA, cvPoint (0, 0));
cvDrawContours (dst_img[2], contours, CV_RGB (255, 0, 0), CV_RGB (0, 255, 0), levels + 1,
1, CV_AA, cvPoint (0, 0));
//=====
imagecontour_1 = cvCloneImage(dst_img[2]);
for (i = 0; i < 3; i++) {
    cvReleaseImage (&dst_img[i]);
}
cvFree (dst_img);
cvReleaseMemStorage (&storage);
//=====

```

```

//===== Tampilkan image 1 =====
if (imagecontour_1 !=0)
{
    if(tampil_1 == 0)
    {
        tampil_1 = cvCreateImage( cvGetSize(imagecontour_1), 8, 3 );
    }
    if(tampil_1 != 0)
    {
        cvReleaseImage( &tampil_1 ); // fungsinya menghapus image jika tidak digunakan
        tampil_1 = cvCreateImage( cvGetSize(imagecontour_1), 8, 3 );
    }
    //=====
    cvCvtColor(imagecontour_1, tampil_1, CV_GRAY2BGR);
    pictureBox_1->Image = gcnew
    System::Drawing::Bitmap(tampil_1->width,tampil_1->height,tampil_1->widthStep,
    System::Drawing::Imaging::PixelFormat::Format24bppRgb,(System::IntPtr)tampil_1-
>imageData);
    pictureBox_1->Refresh();
}

```

Catatan:

Perintah BLOB ANALYSIS diatas dilakukan juga untuk image 2 sampai image 6

```

if (pilih_operasi==1){this->btn_baca_koordinat_Click_1(sender, e);}
    } // akhir proses dari kamera

// proses dari gambar scan
if (status_sumber==0)
{
    int count_lama=0;
    // blob analysis Image scan
    IplImage **dst_img;
    int levels = 0;
    CvMemStorage *storageScan = cvCreateMemStorage (0);
    CvSeq *contoursScan = 0;
    cvClearMemStorage( storageScan );
    levels = 0;
    dst_img = (IplImage **) cvAlloc (sizeof (IplImage *) * 3);
    for (i = 0; i < 3; i++) {
        dst_img[i] = cvCloneImage (erot);};
    int Nc = cvFindContours(erot,storageScan,&contoursScan,sizeof (CvContour),CV_RETR_LIST);
    int batas = 5;
    int tot=0;
    //=====
    StreamWriter^ outFileScan = gcnew StreamWriter ("Data_XY_imageScan.txt");
    //=====
    CvPoint mins[10000],maxs[10000];
    CvPoint minsy[10000],maxsy[10000];
    int n=0,pos=0,ulang=0;
    double buf=0;
    int mak1=1,mak2=0,n1=0,n2=0,i=0;
    //=====
    for (i=0;i<10000;i++){garis_xScan[i]=0;garis_yScan[i]=0;};
    countScan=0;
    for( CvSeq* c=contoursScan; c!=NULL; c=c->h_next )
    {
        tot=c->total;
        if(tot>batas){countScan++;};
        //=====
        int total_yScan[10000], total_xScan[10000];
        int maxx=0,maxys=0,minx=10000,minys=10000;
        if(c->total>mak1){mak1=c->total;n1=n;};
        for( i=0; i<c->total; i=i+1 )
        {
            CvPoint* p = CV_GET_SEQ_ELEM( CvPoint, c, i );
            if(maxx<p->x)
            {
                maxs[n].x=p->x;
                maxs[n].y=p->y;
                maxx=p->x;
            }
        }
    }
}

```

```

        }
        if(minx>p->x)
        {
            mins[n].x=p->x;
            mins[n].y=p->y;
            minx=p->x;
        }
        if(maxys<p->y)
        {
            maxsy[n].x=p->x;
            maxsy[n].y=p->y;
            maxys=p->y;
        }
        if(minys>p->y)
        {
            minsy[n].x=p->x;
            minsy[n].y=p->y;
            minys=p->y;
        }
    }
    if (count_lama != countScan)
    {
        count_lama=countScan;
        rata_xScan[countScan]=(minx+maxx)/2;
        rata_yScan[countScan]=(minys+maxys)/2;
        //==
        outFileScan->Write("Dot ke: ");
        outFileScan->Write(Convert::ToString(countScan));
        outFileScan->Write(" >>> X= ");
        outFileScan->Write(Convert::ToString(rata_xScan[countScan]));
        outFileScan->Write(" , Y= ");
        outFileScan->Write(Convert::ToString(rata_yScan[countScan]));
        outFileScan->Write("\r\n");
    }
    n++;
}
outFileScan->Write("Banyak Dot: ");
outFileScan->Write(Convert::ToString(countScan));
outFileScan->Write("\r\n");
outFileScan->Close();
//=====
cvDrawContours (dst_img[0], contoursScan, CV_RGB (255, 0, 0), CV_RGB (0, 255, 0), levels -
1, 2, CV_AA, cvPoint (0, 0));
cvDrawContours (dst_img[1], contoursScan, CV_RGB (255, 0, 0), CV_RGB (0, 255, 0), levels,
2, CV_AA, cvPoint (0, 0));
cvDrawContours (dst_img[2], contoursScan, CV_RGB (255, 0, 0), CV_RGB (0, 255, 0), levels +
1, 1, CV_AA, cvPoint (0, 0));
//=====
imagecontour_Scan = cvCloneImage(dst_img[2]);
for (i = 0; i < 3; i++) {
    cvReleaseImage (&dst_img[i]);
}
cvFree (dst_img);
cvReleaseMemStorage (&storageScan);
//=====
//===== Tampilkan image =====
if (imagecontour_Scan !=0)
{
    if(tampil_Scan == 0)
    {
        tampil_Scan = cvCreateImage( cvGetSize(imagecontour_Scan), 8, 3 );
    }
    if(tampil_Scan != 0)
    {
        cvReleaseImage( &tampil_Scan ); // fungsinya menghapus image jika tidak digunakan
        tampil_Scan = cvCreateImage( cvGetSize(imagecontour_Scan), 8, 3 );
    }
}
//=====
cvCvtColor(imagecontour_Scan, tampil_Scan, CV_GRAY2BGR);
pictureBox_1->Image = gcnw
System::Drawing::Bitmap(tampil_Scan->width,tampil_Scan->height,tampil_Scan->widthStep,

```

```

System::Drawing::Imaging::PixelFormat::Format24bppRgb, (System::IntPtr)tampil_Scan-
>imageData);
pictureBox_1->Refresh();
}
if (pilih_operasi==1){this->btn_baca_koordinat_Click(sender, e);}
} // akhir proses dari scan
}

// kirim kode siap ke mikro
private: System::Void btn_siap_mikro_Click(System::Object^ sender, System::EventArgs^ e)
{
//
    serialPort1->WriteLine("D"); // Status siap pada mikro diubah ke 1
    Sleep(100);
}

// button segmentasi area
private: System::Void btn_segmentasi_area_Click(System::Object^ sender,
System::EventArgs^ e) {

int awal_baris=1,awal_kolom=1,bb=0,kk=0,cek_x=0,cek_y=0,itung_kode=0,hasil_baca=0;
int acuan_x=0,acuan_y=0;
int scanX_awal=0,scanX_akhir=0,scanY_awal=0,scanY_akhir=0,scan_X=0,scan_Y=0;
int area_X_awal=0,area_X_akhir=0;
int area_Y_awal=0,area_Y_akhir=0;
    int n1=0,m1=0,segmen_X=0,segmen_Y=0;
    int perulangan=0;

    // Proses dari kamera
    if (status_sumber==1)
    {
        // isi image segmentasi 1
        if(segmentasi_1 == 0)
        {
            segmentasi_1 = cvCreateImage( cvGetSize(tampil_1), 8, 3 );
        }
        if(segmentasi_1 != 0)
        {
            cvReleaseImage( &segmentasi_1 );
            segmentasi_1 = cvCreateImage( cvGetSize(tampil_1), 8, 3 );
        }
        segmentasi_1=cvCloneImage(tampil_1);
        //=====
Catatan:
Perintah segmentasi diatas dilakukan juga untuk image 2 sampai image 6

for (perulangan=1; perulangan<=6; perulangan++)
{
    if (perulangan==1){baris=baris1; kolom=kolom1;};
    if (perulangan==2){baris=baris2; kolom=kolom2;};
    if (perulangan==3){baris=baris3; kolom=kolom3;};
    if (perulangan==4){baris=baris4; kolom=kolom4;};
    if (perulangan==5){baris=baris5; kolom=kolom5;};
    if (perulangan==6){baris=baris6; kolom=kolom6;};

    for (bb=awal_baris;bb<baris;bb=bb+3)
    {
        for (kk=awal_kolom;kk<kolom;kk=kk+2)
        {
            itung_kode=0;
            for (cek_y=bb;cek_y<bb+3;cek_y++)
            {
                for (cek_x=kk;cek_x<kk+2;cek_x++)
                {
                    hasil_baca=0;
                    itung_kode++;
                }
            }
            if (perulangan==1) {acuan_x=koordinat_x1[cek_x]; acuan_y=koordinat_y1[cek_y];};
            if (perulangan==2) {acuan_x=koordinat_x2[cek_x]; acuan_y=koordinat_y2[cek_y];};
            if (perulangan==3) {acuan_x=koordinat_x3[cek_x]; acuan_y=koordinat_y3[cek_y];};
            if (perulangan==4) {acuan_x=koordinat_x4[cek_x]; acuan_y=koordinat_y4[cek_y];};

```

```

        if (perulangan==5) {acuan_x=koordinat_x5[cek_x]; acuan_y=koordinat_y5[cek_y];};
        if (perulangan==6) {acuan_x=koordinat_x6[cek_x]; acuan_y=koordinat_y6[cek_y];};
        //=====
        scanX_awal=acuan_x - 3;
        scanX_akhir=acuan_x + 4;
        scanY_awal=acuan_y - 3;
        scanY_akhir=acuan_y + 1;
        //=====
        if (itung_kode==1){area_X_awal=scanX_awal;area_Y_awal=scanY_awal;sudut1.y =
scanY_awal-1;sudut1.x = scanX_awal-1;};
        if (itung_kode==6)
        {
            area_X_akhir=scanX_akhir;
            area_Y_akhir=scanY_akhir;
            sudut2.y = scanY_akhir+3;
            sudut2.x = scanX_akhir+1;
        }
        if (perulangan==1) {cvRectangle(segmentasi_1,sudut2,sudut1,CV_RGB(0,0,255),2,8,0);}
        if (perulangan==2) {cvRectangle(segmentasi_2,sudut2,sudut1,CV_RGB(0,0,255),2,8,0);}
        if (perulangan==3) {cvRectangle(segmentasi_3,sudut2,sudut1,CV_RGB(0,0,255),2,8,0);}
        if (perulangan==4) {cvRectangle(segmentasi_4,sudut2,sudut1,CV_RGB(0,0,255),2,8,0);}
        if (perulangan==5) {cvRectangle(segmentasi_5,sudut2,sudut1,CV_RGB(0,0,255),2,8,0);}
        if (perulangan==6) {cvRectangle(segmentasi_6,sudut2,sudut1,CV_RGB(0,0,255),2,8,0);}

    };
}
}
// segmentasi kecil (sebanyak 40 >> 5 kolom dan 8 baris
segmen_X=4;
segmen_Y=4;
itung_kode=0;
for (m1=0;m1<5;m1++)
{
    sudut1.x = area_X_awal + (segmen_X * m1);
    sudut2.x = area_X_awal + (segmen_X * (m1+1));
    for (n1=0;n1<8;n1++)
    {
        sudut1.y = area_Y_awal + (segmen_Y * n1);
        sudut2.y = area_Y_awal + (segmen_Y * (n1+1));
        if (perulangan==1) {cvRectangle(segmentasi_1,sudut2,sudut1,CV_RGB(255,0,0),1,8,0);}
        if (perulangan==2) {cvRectangle(segmentasi_2,sudut2,sudut1,CV_RGB(255,0,0),1,8,0);}
        if (perulangan==3) {cvRectangle(segmentasi_3,sudut2,sudut1,CV_RGB(255,0,0),1,8,0);}
        if (perulangan==4) {cvRectangle(segmentasi_4,sudut2,sudut1,CV_RGB(255,0,0),1,8,0);}
        if (perulangan==5) {cvRectangle(segmentasi_5,sudut2,sudut1,CV_RGB(255,0,0),1,8,0);}
        if (perulangan==6) {cvRectangle(segmentasi_6,sudut2,sudut1,CV_RGB(255,0,0),1,8,0);}
    }
}
}
}
// tampilkan image 1 di picture box
pictureBox_1->Image = gcnew
System::Drawing::Bitmap(segmentasi_1->width,segmentasi_1->height,segmentasi_1->widthStep,
System::Drawing::Imaging::PixelFormat::Format24bppRgb,(System::IntPtr)segmentasi_1-
>imageData);
pictureBox_1->Refresh();
//=====
// tampilkan image 2 di picture box
pictureBox_2->Image = gcnew
System::Drawing::Bitmap(segmentasi_2->width,segmentasi_2->height,segmentasi_2->widthStep,
System::Drawing::Imaging::PixelFormat::Format24bppRgb,(System::IntPtr)segmentasi_2-
>imageData);
pictureBox_2->Refresh();
//=====
// tampilkan image 3 di picture box
pictureBox_3->Image = gcnew
System::Drawing::Bitmap(segmentasi_3->width,segmentasi_3->height,segmentasi_3->widthStep,
System::Drawing::Imaging::PixelFormat::Format24bppRgb,(System::IntPtr)segmentasi_3-
>imageData);
pictureBox_3->Refresh();
//=====

```

```

// tampilkan image 4 di picture box
pictureBox_4->Image = gcnew
System::Drawing::Bitmap(segmentasi_4->width,segmentasi_4->height,segmentasi_4->widthStep,
System::Drawing::Imaging::PixelFormat::Format24bppRgb,(System::IntPtr)segmentasi_4-
>imageData);
pictureBox_4->Refresh();
//=====
// tampilkan image 5 di picture box
pictureBox_5->Image = gcnew
System::Drawing::Bitmap(segmentasi_5->width,segmentasi_5->height,segmentasi_5->widthStep,
System::Drawing::Imaging::PixelFormat::Format24bppRgb,(System::IntPtr)segmentasi_5-
>imageData);
pictureBox_5->Refresh();
//=====
// tampilkan image 6 di picture box
pictureBox_6->Image = gcnew
System::Drawing::Bitmap(segmentasi_6->width,segmentasi_6->height,segmentasi_6->widthStep,
System::Drawing::Imaging::PixelFormat::Format24bppRgb,(System::IntPtr)segmentasi_6-
>imageData);
pictureBox_6->Refresh();
//=====
if (pilih_operasi==1){this->btn_proses_ANN_Click(sender, e);}
} // akhir proses dari kamera
//=====
// proses segmentasi dari gambar scan
if (status_sumber==0)
{
int tesawal=0;
baris=barisScan;
kolom=kolomScan;
for (bb=awal_baris;bb<baris;bb=bb+3)
{
    for (kk=awal_kolom;kk<kolom;kk=kk+2)
    {
        itung_kode=0;
        for (cek_y=bb;cek_y<bb+3;cek_y++)
        {
            for (cek_x=kk;cek_x<kk+2;cek_x++)
            {
                hasil_baca=0;
                itung_kode++;
                acuan_x=koordinat_xScan[cek_x]; acuan_y=koordinat_yScan[cek_y];
                //=====
                scanX_awal=acuan_x - 3;
                scanX_akhir=acuan_x + 4;
                scanY_awal=acuan_y - 0;
                scanY_akhir=acuan_y + 1;
                //=====
            }
        }
        if (itung_kode==1){area_X_awal=scanX_awal;area_Y_awal=scanY_awal;sudut1.y = scanY_awal-1;sudut1.x = scanX_awal-1;};
        if (itung_kode==6)
        {
            area_X_akhir=scanX_akhir;
            area_Y_akhir=scanY_akhir;
            sudut2.y = scanY_akhir+1;
            sudut2.x = scanX_akhir;
            cvRectangle(tampil_Scan,sudut2,sudut1,CV_RGB(0,0,255),1,8,0);
        };
    };
}
//=====
}

// segmentasi kecil (sebanyak 40 >> 5 kolom dan 8 baris
segmen_X=3;
segmen_Y=2;
itung_kode=0;
for (m1=0;m1<5;m1++)
{
    sudut1.x = area_X_awal + (segmen_X * m1);
    sudut2.x = area_X_awal + (segmen_X * (m1+1));
    for (n1=0;n1<8;n1++)

```

```

        {
            sudut1.y = area_Y_awal + (segmen_Y * n1);
            sudut2.y = area_Y_awal + (segmen_Y * (n1+1));
            if (n1==7) {sudut2.y = sudut2.y +1;};
            cvRectangle(tampil_Scan,sudut2,sudut1,CV_RGB(255,0,0),1,8,0);
        }
    }
}
// tampilkan image
pictureBox_1->Image = gcnew
System::Drawing::Bitmap(tampil_Scan->width,tampil_Scan->height,tampil_Scan->widthStep,
System::Drawing::Image::PixelFormat::Format24bppRgb, (System::IntPtr)tampil_Scan-
>imageData);
pictureBox_1->Refresh();
//=====
} // Akhir proses gambar scan
} // akhir buton segmentasi

// Proses Pengenalan huruf braille dengan ANN
private: System::Void btn_proses_ANN_Click(System::Object^ sender, System::EventArgs^ e)
{
    StreamWriter^ tulis_hasil_1 = gcnew StreamWriter ("hasil_text_1.txt");
    StreamWriter^ tulis_hasil_2 = gcnew StreamWriter ("hasil_text_2.txt");
    StreamWriter^ tulis_hasil_3 = gcnew StreamWriter ("hasil_text_3.txt");
    StreamWriter^ tulis_hasil_4 = gcnew StreamWriter ("hasil_text_4.txt");
    StreamWriter^ tulis_hasil_5 = gcnew StreamWriter ("hasil_text_5.txt");
    StreamWriter^ tulis_hasil_6 = gcnew StreamWriter ("hasil_text_6.txt");
    StreamWriter^ baca_input_seg = gcnew StreamWriter ("input_seg.txt");
    //=====
    richTextBox_hasil->Text="";
    richTextBox_hasil->Text="          INI ADALAH HASIL KONVERSI\r\n";
    textBox_huruf->Text="";
    this->btn_ambil_bobot_ANN_Click_1(sender, e); // ambil bobot untuk proses ANN
    baca_input_seg->Write(" Hasil Pembacaan data input pada segmentasi kecil per satu
    karakter\r\n");
    int
    awal_baris=1,awal_kolom=1,bb=0,kk=0,cek_x=0,cek_y=0,itung_kode=0,hasil_baca=0,acuan_x=0,ac
    uan_y=0;
    int
    scanX_awal=0,scanX_akhir=0,scanY_awal=0,scanY_akhir=0,scan_X=0,scan_Y=0,kode_in[50],jumlah
    baca;
    int
    perulangan=0,m1,n1,m2,segmen_X,segmen_Y,area_X_awal,area_Y_awal,area_X_akhir,area_Y_akhir;
    int bitke0,bitke1,bitke2,bitke3,bitke4,bitke5,itung_huruf=0;

    tanda_angka=0;
    kapital=0;
    for (perulangan=1; perulangan<=6; perulangan++)
    {
        if (perulangan==1){baris=baris1; kolom=kolom1;baca_input_seg->Write("Data Image 1\r\n");};
        if (perulangan==2){baris=baris2; kolom=kolom2;baca_input_seg->Write("Data Image 2\r\n");};
        if (perulangan==3){baris=baris3; kolom=kolom3;baca_input_seg->Write("Data Image 3\r\n");};
        if (perulangan==4){baris=baris4; kolom=kolom4;baca_input_seg->Write("Data Image 4\r\n");};
        if (perulangan==5){baris=baris5; kolom=kolom5;baca_input_seg->Write("Data Image 5\r\n");};
        if (perulangan==6){baris=baris6; kolom=kolom6;baca_input_seg->Write("Data Image 6\r\n");};
        itung_huruf=0;
        for (bb=awal_baris;bb<baris;bb=bb+3)
        {
            for (kk=awal_kolom;kk<kolom;kk=kk+2)
            {
                itung_kode=0;
                for (cek_y=bb;cek_y<bb+3;cek_y++)
                {
                    for (cek_x=kk;cek_x<kk+2;cek_x++)
                    {
                        hasil_baca=0;
                        itung_kode++;
                    }
                }
            }
        }
    }
}

```

```

        if (perulangan==1) {acuan_x=koordinat_x1[cek_x]; acuan_y=koordinat_y1[cek_y];};
        if (perulangan==2) {acuan_x=koordinat_x2[cek_x]; acuan_y=koordinat_y2[cek_y];};
        if (perulangan==3) {acuan_x=koordinat_x3[cek_x]; acuan_y=koordinat_y3[cek_y];};
        if (perulangan==4) {acuan_x=koordinat_x4[cek_x]; acuan_y=koordinat_y4[cek_y];};
        if (perulangan==5) {acuan_x=koordinat_x5[cek_x]; acuan_y=koordinat_y5[cek_y];};
        if (perulangan==6) {acuan_x=koordinat_x6[cek_x]; acuan_y=koordinat_y6[cek_y];};
        //=====
        scanX_awal=acuan_x - 3;
        scanX_akhir=acuan_x + 4;
        scanY_awal=acuan_y - 3;
        scanY_akhir=acuan_y + 1;
        //=====
        if (itung_kode==1){area_X_awal=scanX_awal;area_Y_awal=scanY_awal;sudut1.y = scanY_awal-1;sudut1.x = scanX_awal-1;};
        if (itung_kode==6)
        {
            area_X_akhir=scanX_akhir;
            area_Y_akhir=scanY_akhir;
            sudut2.y = scanY_akhir+3;
            sudut2.x = scanX_akhir+1;
        };
        //=====
    }

    // segmentasi kecil (sebanyak 40 >> 5 kolom dan 8 baris)
    segmen_X=4;
    segmen_Y=4;
    itung_kode=0;
    for (m1=0;m1<5;m1++)
    {
        sudut1.x = area_X_awal + (segmen_X * m1);
        sudut2.x = area_X_awal + (segmen_X * (m1+1));
        for (n1=0;n1<8;n1++)
        {
            sudut1.y = area_Y_awal + (segmen_Y * n1);
            sudut2.y = area_Y_awal + (segmen_Y * (n1+1));
            //=== pembacaan area tiap segmentasi kecil. apakah hitam atau putih
            hitam=0;putih=0;
            hasil_baca=0;
            itung_kode++;
            for (scan_Y=sudut1.y;scan_Y<=sudut2.y;scan_Y++)
            {
                for (scan_X=sudut1.x;scan_X<=sudut2.x;scan_X++)
                {
                    if (perulangan==1)
                    {
                        if (imagecontour_1->imageData[imagecontour_1->widthStep*scan_Y+scan_X*imagecontour_1->nChannels] == 0){hitam++;}
                        else
                        {putih++;};
                    }

                    if (perulangan==2)
                    {
                        if (imagecontour_2->imageData[imagecontour_2->widthStep*scan_Y+scan_X*imagecontour_2->nChannels] == 0){hitam++;}
                        else
                        {putih++;};
                    }

                    if (perulangan==3)
                    {
                        if (imagecontour_3->imageData[imagecontour_3->widthStep*scan_Y+scan_X*imagecontour_3->nChannels] == 0){hitam++;}
                        else
                        {putih++;};
                    }
                    if (perulangan==4)
                    {

```



```

if (imagecontour_4->imageData[imagecontour_4->widthStep*scan_Y+scan_X*imagecontour_4-
>nChannels] == 0){hitam++;}
else
{putih++;};
}

if (perulangan==5)
{
if (imagecontour_5->imageData[imagecontour_5->widthStep*scan_Y+scan_X*imagecontour_5-
>nChannels] == 0){hitam++;}
else
{putih++;};
}

if (perulangan==6)
{
if (imagecontour_6->imageData[imagecontour_6->widthStep*scan_Y+scan_X*imagecontour_6-
>nChannels] == 0){hitam++;}
else
{putih++;};
}; }; // akhir pembacaan area segmentasi kecil
if (hitam>2) {hasil_baca=1;}
//=====
if (itung_kode==1){kode_in[1]=hasil_baca;};
if (itung_kode==2){kode_in[3]=hasil_baca;};
if (itung_kode==3){kode_in[5]=hasil_baca;};
if (itung_kode==4){kode_in[7]=hasil_baca;};
if (itung_kode==5){kode_in[9]=hasil_baca;};
if (itung_kode==6){kode_in[11]=hasil_baca;};
if (itung_kode==7){kode_in[13]=hasil_baca;};
if (itung_kode==8){kode_in[15]=hasil_baca;};
if (itung_kode==9){kode_in[2]=hasil_baca;};
if (itung_kode==10){kode_in[4]=hasil_baca;};
if (itung_kode==11){kode_in[6]=hasil_baca;};
if (itung_kode==12){kode_in[8]=hasil_baca;};
if (itung_kode==13){kode_in[10]=hasil_baca;};
if (itung_kode==14){kode_in[12]=hasil_baca;};
if (itung_kode==15){kode_in[14]=hasil_baca;};
if (itung_kode==16){kode_in[16]=hasil_baca;};
if (itung_kode==17){kode_in[17]=hasil_baca;};
if (itung_kode==18){kode_in[18]=hasil_baca;};
if (itung_kode==19){kode_in[19]=hasil_baca;};
if (itung_kode==20){kode_in[20]=hasil_baca;};
if (itung_kode==21){kode_in[21]=hasil_baca;};
if (itung_kode==22){kode_in[22]=hasil_baca;};
if (itung_kode==23){kode_in[23]=hasil_baca;};
if (itung_kode==24){kode_in[24]=hasil_baca;};
if (itung_kode==25){kode_in[25]=hasil_baca;};
if (itung_kode==26){kode_in[27]=hasil_baca;};
if (itung_kode==27){kode_in[29]=hasil_baca;};
if (itung_kode==28){kode_in[31]=hasil_baca;};
if (itung_kode==29){kode_in[33]=hasil_baca;};
if (itung_kode==30){kode_in[35]=hasil_baca;};
if (itung_kode==31){kode_in[37]=hasil_baca;};
if (itung_kode==32){kode_in[39]=hasil_baca;};
if (itung_kode==33){kode_in[26]=hasil_baca;};
if (itung_kode==34){kode_in[28]=hasil_baca;};
if (itung_kode==35){kode_in[30]=hasil_baca;};
if (itung_kode==36){kode_in[32]=hasil_baca;};
if (itung_kode==37){kode_in[34]=hasil_baca;};
if (itung_kode==38){kode_in[36]=hasil_baca;};
if (itung_kode==39){kode_in[38]=hasil_baca;};
if (itung_kode==40){kode_in[40]=hasil_baca;};
//=====
} // akhir scan n1
} // akhir scan m1

//PROSES A N N
//=====
for (m1 = 1; m1 <= node_hidden1; m1++)
{

```

```

        temp=0;
        for (n1 = 1; n1 <= banyak_input; n1++)
        {
            temp = temp + (WL1[m1][n1] * kode_in[n1]);
        }
        temp = temp + BIAS [1][m1];
        HASIL_LAYER[1][m1] = 1/(1+exp(-alpha * temp));
        Y[1][m1] = HASIL_LAYER[1][m1];
    }
    // y = fungsi sigmoid(sigma w*x) untuk hidden layer ke-2
    for (m1 = 1; m1 <= node_hidden2; m1++)
    {
        temp = 0;
        for (n1 = 1; n1 <= node_hidden1; n1++)
        {
            temp = temp + (WL2[m1][n1] * Y[1][n1]);
        }
        temp = temp + BIAS [2][m1];
        HASIL_LAYER[2][m1] = 1/(1+exp(-alpha * temp));
        Y[2][m1] = HASIL_LAYER[2][m1];
    }

    // y = fungsi sigmoid(sigma w*x) untuk hidden2 layer output
    for (m1 = 1; m1 <= node_output; m1++)
    {
        temp = 0;
        for (n1 = 1; n1 <= node_hidden2; n1++)
        {
            temp = temp + (WL3[m1][n1] * Y[2][n1]);
        }
        temp = temp + BIAS [3][m1];
        HASIL_LAYER[3][m1] = 1/(1+exp(-alpha * temp));
        Y[3][m1] = HASIL_LAYER[3][m1];
    }
    //=====
    bitke0=0;bitke1=0;bitke2=0;bitke3=0;bitke4=0;bitke5=0;
    if (Y[3][1] >= thresx){bitke0=1;};
    if (Y[3][2] >= thresx){bitke1=1;};
    if (Y[3][3] >= thresx){bitke2=1;};
    if (Y[3][4] >= thresx){bitke3=1;};
    if (Y[3][5] >= thresx){bitke4=1;};
    if (Y[3][6] >= thresx){bitke5=1;};
    //=====
    jumlahbaca=0;
    if (bitke0==1){jumlahbaca=jumlahbaca+1;};
    if (bitke1==1){jumlahbaca=jumlahbaca+2;};
    if (bitke2==1){jumlahbaca=jumlahbaca+4;};
    if (bitke3==1){jumlahbaca=jumlahbaca+8;};
    if (bitke4==1){jumlahbaca=jumlahbaca+16;};
    if (bitke5==1){jumlahbaca=jumlahbaca+32;};
    //=====
    // Spasi
    if (jumlahbaca==0){textBox_huruf->Text=" ";tanda_angka=0;};
    // Huruf A
    if (jumlahbaca==1){if (tanda_angka==1){textBox_huruf->Text="1";};
    else {if (kapital==1){textBox_huruf->Text="A";kapital=0;};
    else {textBox_huruf->Text="a";};
    };};
    // Huruf B
    if (jumlahbaca==2){
    if (tanda_angka==1){textBox_huruf->Text="2";};
    else {if (kapital==1){textBox_huruf->Text="B";kapital=0;};
    else {textBox_huruf->Text="b";};} }
    // Huruf C
    if (jumlahbaca==3){
    if (tanda_angka==1){textBox_huruf->Text="3";};
    else {if (kapital==1){textBox_huruf->Text="C";kapital=0;};
    else {textBox_huruf->Text="c";}; } }
    // Huruf D
    if (jumlahbaca==4){
    if (tanda_angka==1){textBox_huruf->Text="4";};
    else {if (kapital==1){textBox_huruf->Text="D";kapital=0;};

```

```

else {textBox_huruf->Text="d";}} }
// Huruf E
if (jumlahbaca==5){
if (tanda_angka==1){textBox_huruf->Text="5";}
else {if (kapital==1){textBox_huruf->Text="E";kapital=0;}
else {textBox_huruf->Text="e";}} }
// Huruf F
if (jumlahbaca==6) {
if (tanda_angka==1){textBox_huruf->Text="6";}
else {if (kapital==1){textBox_huruf->Text="F";kapital=0;}
else {textBox_huruf->Text="f";} } }
// Huruf G
if (jumlahbaca==7) {
if (tanda_angka==1){textBox_huruf->Text="7";}
else {if (kapital==1){textBox_huruf->Text="G";kapital=0;}
else {textBox_huruf->Text="g";} } }
// Huruf H
if (jumlahbaca==8) {
if (tanda_angka==1){textBox_huruf->Text="8";}
else {if (kapital==1){textBox_huruf->Text="H";kapital=0;}
else {textBox_huruf->Text="h";} } }
// Huruf I
if (jumlahbaca==9){
if (tanda_angka==1){textBox_huruf->Text="9";}
else {if (kapital==1){textBox_huruf->Text="I";kapital=0;}
else {textBox_huruf->Text="i";} } }
// Huruf J
if (jumlahbaca==10){
if (tanda_angka==1){textBox_huruf->Text="0";}
else {if (kapital==1){textBox_huruf->Text="J";kapital=0;}
else {textBox_huruf->Text="j";} } }
// Huruf K
if (jumlahbaca==11){
if (kapital==1){textBox_huruf->Text="K";kapital=0;}
else {textBox_huruf->Text="k";} } }
// Huruf L
if (jumlahbaca==12){
if (kapital==1){textBox_huruf->Text="L";kapital=0;}
else {textBox_huruf->Text="l";} } }
// Huruf M
if (jumlahbaca==13){
if (kapital==1){textBox_huruf->Text="M";kapital=0;}
else {textBox_huruf->Text="m";} } }
// Huruf N
if (jumlahbaca==14){
if (kapital==1){textBox_huruf->Text="N";kapital=0;}
else {textBox_huruf->Text="n";} } }
// Huruf O
if (jumlahbaca==15) {
if (kapital==1){textBox_huruf->Text="O";kapital=0;}
else {textBox_huruf->Text="o";} } }
// Huruf P
if (jumlahbaca==16) {
if (kapital==1){textBox_huruf->Text="P";kapital=0;}
else {textBox_huruf->Text="p";} } }
// Huruf Q
if (jumlahbaca==17) {
if (kapital==1){textBox_huruf->Text="Q";kapital=0;}
else {textBox_huruf->Text="q";} } }
// Huruf R
if (jumlahbaca==18){
if (kapital==1){textBox_huruf->Text="R";kapital=0;}
else {textBox_huruf->Text="r";} } }
// Huruf S
if (jumlahbaca==19) {
if (kapital==1){textBox_huruf->Text="S";kapital=0;}
else {textBox_huruf->Text="s";}} }
// Huruf T
if (jumlahbaca==20) {
if (kapital==1){textBox_huruf->Text="T";kapital=0;}
else {textBox_huruf->Text="t";} } }

```

```

// Huruf U
if (jumlahbaca==21) {
if (kapital==1){textBox_huruf->Text="U";kapital=0;}
else {textBox_huruf->Text="u";} }
// Huruf V
if (jumlahbaca==22) {
if (kapital==1){textBox_huruf->Text="V";kapital=0;}
else {textBox_huruf->Text="v";} }
// Huruf W
if (jumlahbaca==23) {
if (kapital==1){textBox_huruf->Text="W";kapital=0;}
else {textBox_huruf->Text="w";} }
// Huruf X
if (jumlahbaca==24){
if (kapital==1){textBox_huruf->Text="X";kapital=0;}
else {textBox_huruf->Text="x";} }

// Huruf Y
if (jumlahbaca==25) {
if (kapital==1){textBox_huruf->Text="Y";kapital=0;}
else {textBox_huruf->Text="y";} }
// Huruf Z
if (jumlahbaca==26) {
if (kapital==1){textBox_huruf->Text="Z";kapital=0;}
else {textBox_huruf->Text="z";} }
// Kapital
if (jumlahbaca==27){kapital=1;}
// tanda angka
if (jumlahbaca==28){tanda_angka=1;}
// Titik (.)
if (jumlahbaca==29){textBox_huruf->Text=".";}
// Tanda tanya (?)
if (jumlahbaca==30){textBox_huruf->Text="?";tanda_angka=0;kapital=0;}
// Koma
if (jumlahbaca==31){textBox_huruf->Text=",";}
// Tanda Seru
if (jumlahbaca==32){textBox_huruf->Text="!";}
// Tanda Strip -
if (jumlahbaca==33){textBox_huruf->Text="-";}
// Karakter lainnya
if (jumlahbaca>33){textBox_huruf->Text=" ";} // diisi spasi

cvWaitKey(50);
textBox_tampung->Text=textBox_tampung->Text+textBox_huruf->Text;
if(textBox_huruf->Text==" "){textBox_huruf->Text=" "+textBox_huruf->Text;};
if(textBox_huruf->Text==","){textBox_huruf->Text=textBox_huruf->Text+",";};
if(textBox_huruf->Text=="."){textBox_huruf->Text=" "+textBox_huruf->Text;};
textBox_huruf->Text="";
//=====
hasil_baca=0;
itung_kode=0;
} // akhir scan kolom
if (perulangan==1) {tulisan_hasil_1->Write(textBox_tampung->Text+"\r\n");}
if (perulangan==2) {tulisan_hasil_2->Write(textBox_tampung->Text+"\r\n");}
if (perulangan==3) {tulisan_hasil_3->Write(textBox_tampung->Text+"\r\n");}
if (perulangan==4) {tulisan_hasil_4->Write(textBox_tampung->Text+"\r\n");}
if (perulangan==5) {tulisan_hasil_5->Write(textBox_tampung->Text+"\r\n");}
if (perulangan==6) {tulisan_hasil_6->Write(textBox_tampung->Text+"\r\n");}
//=====
textBox_tampung->Text="";
} // akhir scan baris
}; // akhir perulangan

tulisan_hasil_1->Close();
tulisan_hasil_2->Close();
tulisan_hasil_3->Close();
tulisan_hasil_4->Close();
tulisan_hasil_5->Close();
tulisan_hasil_6->Close();
//=====

```

```

// Ambil data teks hasil pembacaan
// dan digabungkan menjadi satu
// di tampilkan di richTextBox_hasil
// variabel tampung hasil teks 1 per baris
System::String^ hasil_teks_11;
System::String^ hasil_teks_12;
System::String^ hasil_teks_13;
System::String^ hasil_teks_14;
System::String^ hasil_teks_15;
System::String^ hasil_teks_16;
System::String^ hasil_teks_17;
System::String^ hasil_teks_18;
System::String^ hasil_teks_19;

// variabel tampung hasil teks 2 per baris
System::String^ hasil_teks_21;
System::String^ hasil_teks_22;
System::String^ hasil_teks_23;
System::String^ hasil_teks_24;
System::String^ hasil_teks_25;
System::String^ hasil_teks_26;
System::String^ hasil_teks_27;
System::String^ hasil_teks_28;
System::String^ hasil_teks_29;

// variabel tampung hasil teks 3 per baris
System::String^ hasil_teks_31;
System::String^ hasil_teks_32;
System::String^ hasil_teks_33;
System::String^ hasil_teks_34;
System::String^ hasil_teks_35;
System::String^ hasil_teks_36;
System::String^ hasil_teks_37;
System::String^ hasil_teks_38;
System::String^ hasil_teks_39;

// variabel tampung hasil teks 4 per baris
System::String^ hasil_teks_41;
System::String^ hasil_teks_42;
System::String^ hasil_teks_43;
System::String^ hasil_teks_44;
System::String^ hasil_teks_45;
System::String^ hasil_teks_46;
System::String^ hasil_teks_47;
System::String^ hasil_teks_48;
System::String^ hasil_teks_49;

// variabel tampung hasil teks 5 per baris
System::String^ hasil_teks_51;
System::String^ hasil_teks_52;
System::String^ hasil_teks_53;
System::String^ hasil_teks_54;
System::String^ hasil_teks_55;
System::String^ hasil_teks_56;
System::String^ hasil_teks_57;
System::String^ hasil_teks_58;
System::String^ hasil_teks_59;

// variabel tampung hasil teks 6 per baris
System::String^ hasil_teks_61;
System::String^ hasil_teks_62;
System::String^ hasil_teks_63;
System::String^ hasil_teks_64;
System::String^ hasil_teks_65;
System::String^ hasil_teks_66;
System::String^ hasil_teks_67;
System::String^ hasil_teks_68;
System::String^ hasil_teks_69;

StreamReader^ baca_file1 = gcnew StreamReader("hasil_text_1.txt");
StreamReader^ baca_file2 = gcnew StreamReader("hasil_text_2.txt");

```

```

StreamReader^ baca_file3 = gcnew StreamReader("hasil_text_3.txt");
StreamReader^ baca_file4 = gcnew StreamReader("hasil_text_4.txt");
StreamReader^ baca_file5 = gcnew StreamReader("hasil_text_5.txt");
StreamReader^ baca_file6 = gcnew StreamReader("hasil_text_6.txt");
//=====
hasil_teks_11 = baca_file1->ReadLine();
hasil_teks_12 = baca_file1->ReadLine();
hasil_teks_13 = baca_file1->ReadLine();
hasil_teks_14 = baca_file1->ReadLine();
hasil_teks_15 = baca_file1->ReadLine();
hasil_teks_16 = baca_file1->ReadLine();
hasil_teks_17 = baca_file1->ReadLine();
hasil_teks_18 = baca_file1->ReadLine();
hasil_teks_19 = baca_file1->ReadLine();
baca_file1->Close();
//=====
hasil_teks_21 = baca_file2->ReadLine();
hasil_teks_22 = baca_file2->ReadLine();
hasil_teks_23 = baca_file2->ReadLine();
hasil_teks_24 = baca_file2->ReadLine();
hasil_teks_25 = baca_file2->ReadLine();
hasil_teks_26 = baca_file2->ReadLine();
hasil_teks_27 = baca_file2->ReadLine();
hasil_teks_28 = baca_file2->ReadLine();
hasil_teks_29 = baca_file2->ReadLine();
baca_file2->Close();

//=====
hasil_teks_31 = baca_file3->ReadLine();
hasil_teks_32 = baca_file3->ReadLine();
hasil_teks_33 = baca_file3->ReadLine();
hasil_teks_34 = baca_file3->ReadLine();
hasil_teks_35 = baca_file3->ReadLine();
hasil_teks_36 = baca_file3->ReadLine();
hasil_teks_37 = baca_file3->ReadLine();
hasil_teks_38 = baca_file3->ReadLine();
hasil_teks_39 = baca_file3->ReadLine();
baca_file3->Close();

//=====
hasil_teks_41 = baca_file4->ReadLine();
hasil_teks_42 = baca_file4->ReadLine();
hasil_teks_43 = baca_file4->ReadLine();
hasil_teks_44 = baca_file4->ReadLine();
hasil_teks_45 = baca_file4->ReadLine();
hasil_teks_46 = baca_file4->ReadLine();
hasil_teks_47 = baca_file4->ReadLine();
hasil_teks_48 = baca_file4->ReadLine();
hasil_teks_49 = baca_file4->ReadLine();
baca_file4->Close();

//=====
hasil_teks_51 = baca_file5->ReadLine();
hasil_teks_52 = baca_file5->ReadLine();
hasil_teks_53 = baca_file5->ReadLine();
hasil_teks_54 = baca_file5->ReadLine();
hasil_teks_55 = baca_file5->ReadLine();
hasil_teks_56 = baca_file5->ReadLine();
hasil_teks_57 = baca_file5->ReadLine();
hasil_teks_58 = baca_file5->ReadLine();
hasil_teks_59 = baca_file5->ReadLine();
baca_file5->Close();

//=====
hasil_teks_61 = baca_file6->ReadLine();
hasil_teks_62 = baca_file6->ReadLine();
hasil_teks_63 = baca_file6->ReadLine();
hasil_teks_64 = baca_file6->ReadLine();
hasil_teks_65 = baca_file6->ReadLine();
hasil_teks_66 = baca_file6->ReadLine();
hasil_teks_67 = baca_file6->ReadLine();

```

```

hasil_teks_68 = baca_file6->ReadLine();
hasil_teks_69 = baca_file6->ReadLine();
baca_file6->Close();

//=====
richTextBox_hasil->Text=richTextBox_hasil->Text+hasil_teks_11+hasil_teks_21+"\r\n";
richTextBox_hasil->Text=richTextBox_hasil->Text+hasil_teks_12+hasil_teks_22+"\r\n";
richTextBox_hasil->Text=richTextBox_hasil->Text+hasil_teks_13+hasil_teks_23+"\r\n";
richTextBox_hasil->Text=richTextBox_hasil->Text+hasil_teks_14+hasil_teks_24+"\r\n";
richTextBox_hasil->Text=richTextBox_hasil->Text+hasil_teks_15+hasil_teks_25+"\r\n";
richTextBox_hasil->Text=richTextBox_hasil->Text+hasil_teks_16+hasil_teks_26+"\r\n";
richTextBox_hasil->Text=richTextBox_hasil->Text+hasil_teks_17+hasil_teks_27+"\r\n";
richTextBox_hasil->Text=richTextBox_hasil->Text+hasil_teks_18+hasil_teks_28+"\r\n";
//===

richTextBox_hasil->Text=richTextBox_hasil->Text+hasil_teks_31+hasil_teks_29+"\r\n";
richTextBox_hasil->Text=richTextBox_hasil->Text+hasil_teks_32+hasil_teks_41+"\r\n";
richTextBox_hasil->Text=richTextBox_hasil->Text+hasil_teks_33+hasil_teks_42+"\r\n";
richTextBox_hasil->Text=richTextBox_hasil->Text+hasil_teks_34+hasil_teks_43+"\r\n";
richTextBox_hasil->Text=richTextBox_hasil->Text+hasil_teks_35+hasil_teks_44+"\r\n";
richTextBox_hasil->Text=richTextBox_hasil->Text+hasil_teks_36+hasil_teks_45+"\r\n";
richTextBox_hasil->Text=richTextBox_hasil->Text+hasil_teks_37+hasil_teks_46+"\r\n";
richTextBox_hasil->Text=richTextBox_hasil->Text+hasil_teks_38+hasil_teks_47+"\r\n";
richTextBox_hasil->Text=richTextBox_hasil->Text+hasil_teks_39+hasil_teks_48+"\r\n";
//==

richTextBox_hasil->Text=richTextBox_hasil->Text+hasil_teks_51+hasil_teks_49+"\r\n";
richTextBox_hasil->Text=richTextBox_hasil->Text+hasil_teks_52+hasil_teks_61+"\r\n";
richTextBox_hasil->Text=richTextBox_hasil->Text+hasil_teks_53+hasil_teks_62+"\r\n";
richTextBox_hasil->Text=richTextBox_hasil->Text+hasil_teks_54+hasil_teks_63+"\r\n";
richTextBox_hasil->Text=richTextBox_hasil->Text+hasil_teks_55+hasil_teks_64+"\r\n";
richTextBox_hasil->Text=richTextBox_hasil->Text+hasil_teks_56+hasil_teks_65+"\r\n";
richTextBox_hasil->Text=richTextBox_hasil->Text+hasil_teks_57+hasil_teks_66+"\r\n";
richTextBox_hasil->Text=richTextBox_hasil->Text+hasil_teks_58+hasil_teks_67+"\r\n";
richTextBox_hasil->Text=richTextBox_hasil->Text+hasil_teks_59+hasil_teks_68+"\r\n";
//=====
tabControl1->SelectTab("tabPage5");
if (pilih_operasi==1) {this->btn_siap_mikro_Click_1(sender, e);};

} // Akhir button proses ANN

// button exit
private: System::Void button_exit_Click(System::Object^ sender, System::EventArgs^ e) {

    Close();

}

// button kirim siap ke mikro
private: System::Void btn_siap_mikro_Click_1(System::Object^ sender, System::EventArgs^ e) {

    if (status_sumber==1)

    {

        serialPort1->WriteLine("D"); // Status siap pada mikro diubah ke 1
        Sleep(100);

    }

    label_keterangan->Text="Ready";

}

// button ambil Bobot untuk Proses ANN
private: System::Void btn_ambil_bobot_ANN_Click_1(System::Object^ sender,
System::EventArgs^ e) {
int jjj=0, kkk=0;
System::String^ bantu;
StreamReader^ baca_file = gcnew StreamReader("File_Bobot_ANN.txt");
//=====
// Urutan: Iterasi, Beta, Alpha, node Input, Node Hidden1, Node Hidden2, Node Output,
// Bobot L1 L2 L3, dan Bias 1 2 3
bantu = baca_file->ReadLine(); // keterangan atas
bantu = baca_file->ReadLine(); // keterangan urutan
bantu = baca_file->ReadLine(); // iterasi
bantu = baca_file->ReadLine(); // beta

```

```

bantu = baca_file->ReadLine(); // alpha
alpha = double::Parse(bantu);
bantu = baca_file->ReadLine(); // data input
banyak_input = int::Parse(bantu);
bantu = baca_file->ReadLine(); // node hidden1
node_hidden1 = int::Parse(bantu);
bantu = baca_file->ReadLine(); // node hidden2
node_hidden2 = int::Parse(bantu);
bantu = baca_file->ReadLine(); // node output
node_output = int::Parse(bantu);
bantu = baca_file->ReadLine(); // keterangan ==== Weight layer 1 =====
//=====
for (jjj=1; jjj<=node_hidden1; jjj++)
{
    for (kkk=1; kkk<=banyak_input; kkk++)
    {
        bantu = baca_file->ReadLine(); // ambil bobot
        WL1[jjj][kkk] = double::Parse(bantu);
    }
}

bantu = baca_file->ReadLine(); // keterangan ==== Weight layer 2 =====
//=====
for (jjj=1; jjj<=node_hidden2; jjj++)
{
    for (kkk=1; kkk<=node_hidden1; kkk++)
    {
        bantu = baca_file->ReadLine(); // ambil bobot
        WL2[jjj][kkk] = double::Parse(bantu);
    }
}

bantu = baca_file->ReadLine(); // keterangan ==== Weight layer 3 =====
//=====
for (jjj=1; jjj<=node_output; jjj++)
{
    for (kkk=1; kkk<=node_hidden2; kkk++)
    {
        bantu = baca_file->ReadLine(); // ambil bobot
        WL3[jjj][kkk] = double::Parse(bantu);
    }
}

bantu = baca_file->ReadLine(); // keterangan ==== Bias layer 1 =====
//=====
for (jjj=1; jjj<=node_hidden1; jjj++)
{
    bantu = baca_file->ReadLine(); // ambil BIAS 1
    BIAS [1][jjj] = double::Parse(bantu);
}

bantu = baca_file->ReadLine(); // keterangan ==== Bias layer 2 =====
//=====
for (jjj=1; jjj<=node_hidden2; jjj++)
{
    bantu = baca_file->ReadLine(); // ambil BIAS 2
    BIAS [2][jjj] = double::Parse(bantu);
}

bantu = baca_file->ReadLine(); // keterangan ==== Bias layer 3 =====
//=====
for (jjj=1; jjj<=node_output; jjj++)
{
    bantu = baca_file->ReadLine(); // ambil BIAS 3
    BIAS [3][jjj] = double::Parse(bantu);
}
//=====
baca_file->Close();
} // akhir proses ambil bobot

// baca koordinat baris dan kolom
private: System::Void btn_baca_koordinat_Click_1(System::Object^ sender,
System::EventArgs^ e) {

```



```

// Proses dari kamera
if (status_sumber==1)
{
    // Proses Image 1
    int geser_x1=0,geser_y1=0;
    int dekat_x1=0,dekat_y1=0;
    if(titik_contour_1 == 0)
    {
        titik_contour_1 = cvCreateImage( cvGetSize(tampil_1), 8, 3 );
    }
    if(titik_contour_1 != 0)
    {
        cvReleaseImage( &titik_contour_1 ); // fungsinya menghapus image jika tidak digunakan
        titik_contour_1 = cvCreateImage( cvGetSize(tampil_1), 8, 3 );
    }
    titik_contour_1=cvCloneImage(tampil_1);
    titik_pusat1.x=1000;
    titik_pusat1.y=1000;
    for (tt=1;tt<=count1;tt++)
    {
        // memberi tanda titik merah pada tiap dot
        tanda_merah1.x=rata_x1[tt];
        tanda_merah1.y=rata_y1[tt];
        cvCircle( titik_contour_1,tanda_merah1 , 1, CV_RGB( 255, 0, 0 ), 2,8,0 );
        // mencari start X dan Start Y
        if (rata_x1[tt]<titik_pusat1.x){titik_pusat1.x=rata_x1[tt];};
        if (rata_y1[tt]<titik_pusat1.y){titik_pusat1.y=rata_y1[tt];};
        // tampilkan image 1 di picture box
        pictureBox_1->Image = gcnw
        System::Drawing::Bitmap(titik_contour_1->width,titik_contour_1-
>height,titik_contour_1->widthStep,
        System::Drawing::Image::PixelFormat::Format24bppRgb,(System::IntPtr)titik_contou
r_1->imageData);
        pictureBox_1->Refresh();
        //=====
    }
    startx1=titik_pusat1.x;
    starty1=titik_pusat1.y;
    // mencari nilai x rata rata pada tiap baris
    int selisihx1=0, itungkux1=0, jumlahx1=0, garis_xx1=0, xxx1=0, ganjilx1=0, awalanx1=0;
    garis_xx1=startx1;
    ganjilx1=0;
    kolom1=0;
    for (xxx1=startx1;xxx1<titik_contour_1->width;xxx1=xxx1+1)
    {
        if (awalanx1==0)
        {
            awalanx1=1;
        }
        else
        {
            if (ganjilx1==0)
            {
                ganjilx1=1;
                xxx1=garis_xx1+12;
            }
            else
            {
                ganjilx1=0;
                xxx1=garis_xx1+18;
            }
        }
    }
    itungkux1=0;
    jumlahx1=0;
    for (tt=1;tt<=count1;tt++)
    {
        selisihx1=xxx1-rata_x1[tt];
        if (selisihx1 <5 && selisihx1 >-5){itungkux1++;jumlahx1=jumlahx1+rata_x1[tt];};
    }
    if (itungkux1==0){itungkux1=1;};
    if (jumlahx1!=0)

```

```

{
    garis_xx1=jumlahx1/itungkux1;
    kolom1++;
    koordinat_x1[kolom1]=garis_xx1;
    xxx1=garis_xx1;
}
if (jumlahx1==0)
{
    kolom1++;
    koordinat_x1[kolom1]=xxx1;
    garis_xx1=xxx1;
}

sudut1.x=garis_xx1;
sudut1.y=0;
sudut2.x=garis_xx1;
sudut2.y=titik_contour_1->height;
cvLine(titik_contour_1,sudut1,sudut2, CV_RGB( 255, 0, 0 ),1,8); // gambar garis x
pada rata-rata titik tengah dot
}
//=====
// mencari nilai y rata rata pada tiap baris
int selisihy1=0, itungkuy1=0, jumlahy1=0, garis_yy1=0, yyy1=0, ganjily1=0, awalan1=0;
ganjily1=0;
awalan1=0;
baris1=0;
for (yyy1=starty1;yyy1<titik_contour_1->height;yyy1=yyy1)
{
    if (ganjily1==0){garis_yy1=yyy1;}
    if (ganjily1==1){garis_yy1=garis_yy1+12;}
    if (ganjily1==2){garis_yy1=garis_yy1+12;}
    if (ganjily1==3){garis_yy1=garis_yy1+24;}
    ganjily1++;
    if (ganjily1==4){ganjily1=1;};
    itungkuy1=0;
    jumlahy1=0;
for (tt=0;tt<=count1;tt++)
{
    selisihy1=garis_yy1-rata_y1[tt];
    if (selisihy1 < 5 && selisihy1 > -5){itungkuy1++;jumlahy1=jumlahy1+rata_y1[tt];};
}
if (itungkuy1==0){itungkuy1=1;};
if (jumlahy1!=0)
{
    garis_yy1=jumlahy1/itungkuy1;
    baris1++;
    koordinat_y1[baris1]=garis_yy1;
    yyy1=garis_yy1;
}
if (jumlahy1==0)
{
    baris1++;
    garis_yy1=garis_yy1+1;
    koordinat_y1[baris1]=garis_yy1;
    yyy1=garis_yy1;
}

sudut1.x=1;
sudut1.y=garis_yy1;
sudut2.x=titik_contour_1->width;
sudut2.y=garis_yy1;
cvLine(titik_contour_1,sudut1,sudut2, CV_RGB( 255, 0, 0 ),1,8); // gambar garis x pada
rata-rata titik tengah dot
}
// tampilkan image 1 di picture box
pictureBox_1->Image = gcnew
System::Drawing::Bitmap(titik_contour_1->width,titik_contour_1->height,titik_contour_1-
>widthStep,
System::Drawing::Imaging::PixelFormat::Format24bppRgb,(System::IntPtr)titik_contour_1-
>imageData);
pictureBox_1->Refresh();
//=====
Catatan:

```

Perintah mencari koordinat x dan y diatas dilakukan juga untuk image 2 sampai image 6

```

if (pilih_operasi==1){this->btn_segmentasi_area_Click(sender, e);}
} // Akhir proses dari kamera
//=====================================================
// Proses dari gambar scan
if (status_sumber==0)
{
    // Proses Image Scan
    int geser_xScan=0,geser_yScan=0;
    int dekat_xScan=0,dekat_yScan=0;
    int tambah_x1=8, tambah_x2=12;
    int tambah_y1=8, tambah_y2=8, tambah_y3=14;
    //===
    if(titik_contour_Scan == 0)
    {
        titik_contour_Scan = cvCreateImage( cvGetSize(tampil_Scan), 8, 3 );
    }
    if(titik_contour_Scan != 0)
    {
        cvReleaseImage( &titik_contour_Scan ); // menghapus image jika tidak digunakan
        titik_contour_Scan = cvCreateImage( cvGetSize(tampil_Scan), 8, 3 );
    }
    titik_contour_Scan=cvCloneImage(tampil_Scan);
    titik_pusatScan.x=10000;
    titik_pusatScan.y=10000;
    for (tt=1;tt<=countScan;tt++)
    {
        // memberi tanda titik merah pada tiap dot
        tanda_merahScan.x=rata_xScan[tt];
        tanda_merahScan.y=rata_yScan[tt];
        cvCircle( titik_contour_Scan, tanda_merahScan, 1, CV_RGB( 255, 0, 0 ), 1,8,0 );
        // mencari start X dan Start Y
        if (rata_xScan[tt]<titik_pusatScan.x){titik_pusatScan.x=rata_xScan[tt];};
        if (rata_yScan[tt]<titik_pusatScan.y){titik_pusatScan.y=rata_yScan[tt];};
    }

    startxScan=titik_pusatScan.x;
    startyScan=titik_pusatScan.y;
    cvCircle( titik_contour_Scan, titik_pusatScan, 1, CV_RGB( 0, 0, 255 ), 2,8,0 );
    // mencari nilai x rata rata pada tiap baris
    int selisihxScan=0, itungkuxScan=0, jumlahxScan=0, garis_xxScan=0, xxxScan=0,
    ganjilxScan=0, awalanxScan=0;
    garis_xxScan=startxScan;
    ganjilxScan=0;
    kolomScan=0;
    for (xxxScan=startxScan;xxxScan<titik_contour_Scan->width;xxxScan=xxxScan)
    {
        if (awalanxScan==0)
        {
            awalanxScan=1;
        }
        else
        {
            if (ganjilxScan==0)
            {
                ganjilxScan=1;
                xxxScan=garis_xxScan+tambah_x1;
            }
            else
            {
                ganjilxScan=0;
                xxxScan=garis_xxScan+tambah_x2;
            }
        }
    }
    itungkuxScan=0;
    jumlahxScan=0;
    for (tt=1;tt<=countScan;tt++)
    {
        selisihxScan=xxxScan-rata_xScan[tt];
        if (selisihxScan <5 && selisihxScan >-
5){itungkuxScan++;jumlahxScan=jumlahxScan+rata_xScan[tt];};

```

```

}
if (itungkuxScan==0){itungkuxScan=1;};
if (jumlahxScan!=0)
{
    garis_xxScan=jumlahxScan/itungkuxScan;
    kolomScan++;
    koordinat_xScan[kolomScan]=garis_xxScan;
    xxxScan=garis_xxScan;
}

if (jumlahxScan==0)
{
    kolomScan++;
    koordinat_xScan[kolomScan]=xxxScan;
    garis_xxScan=xxxScan;
}
sudut1.x=garis_xxScan;
sudut1.y=1;
sudut2.x=garis_xxScan;
sudut2.y=titik_contour_Scan->height;
cvLine(titik_contour_Scan,sudut1,sudut2, CV_RGB( 255, 0, 0 ),1,8); // gambar garis x pada
rata-rata titik tengah dot
}
//=====
// mencari nilai y rata rata pada tiap baris
int selisihyScan=0, itungkuyScan=0, jumlahyScan=0, garis_yyScan=0, yyyScan=0,
ganjilyScan=0, awalanScan=0;
ganjilyScan=0;
awalanScan=0;
barisScan=0;
for (yyyScan=startyScan;yyyScan<titik_contour_Scan->height;yyyScan=yyyScan)
{
    if (ganjilyScan==0){garis_yyScan=yyyScan;}
    if (ganjilyScan==1){garis_yyScan=garis_yyScan+tambah_y1;}
    if (ganjilyScan==2){garis_yyScan=garis_yyScan+tambah_y2;}
    if (ganjilyScan==3){garis_yyScan=garis_yyScan+tambah_y3;}
    ganjilyScan++;
    if (ganjilyScan==4){ganjilyScan=1;};
    itungkuyScan=0;
    jumlahyScan=0;
    for (tt=1;tt<=countScan;tt++)
    {
        selisihyScan=garis_yyScan-rata_yScan[tt];
        if (selisihyScan < 5 && selisihyScan > -
5){itungkuyScan++;jumlahyScan=jumlahyScan+rata_yScan[tt];};
if (itungkuyScan==0){itungkuyScan=1;};
if (jumlahyScan!=0)
{
    garis_yyScan=jumlahyScan/itungkuyScan;
    barisScan++;
    koordinat_yScan[barisScan]=garis_yyScan;
    yyyScan=garis_yyScan;
}
if (jumlahyScan==0)
{
    barisScan++;
    garis_yyScan=garis_yyScan+1;
    koordinat_yScan[barisScan]=garis_yyScan;
    yyyScan=garis_yyScan;
}

    sudut1.x=1;
    sudut1.y=garis_yyScan;
    sudut2.x=titik_contour_Scan->width;
    sudut2.y=garis_yyScan;
    cvLine(titik_contour_Scan, sudut1,sudut2, CV_RGB( 255, 0, 0 ),1,8); // gambar
garis x pada rata-rata titik tengah dot
}
// tampilkan image
pictureBox_1->Image = gcnew
System::Drawing::Bitmap(titik_contour_Scan->width,titik_contour_Scan-
>height,titik_contour_Scan->widthStep,

```

```

System::Drawing::Imaging::PixelFormat::Format24bppRgb, (System::IntPtr)titik_contour_Scan-
>imageData);
pictureBox_1->Refresh();
//=====
if (pilih_operasi==1){this->button_baca_area_Click(sender, e);}
} // akhir proses dari scan
}

// button ambil gambar hasil scan
private: System::Void button2_Click(System::Object^ sender, System::EventArgs^ e) {
if(openFileDialog1->ShowDialog() == System::Windows::Forms::DialogResult::OK)
{
    tabControl1->SelectTab("tabPage10");
    origimage = cvLoadImage((char*)(void*)Marshal::StringToHGlobalAnsi(openFileDialog1-
>FileName), 1);
    //=====
    pictureBox_1->Image = gcnew
    System::Drawing::Bitmap(origimage->width, origimage->height, origimage->widthStep,
    System::Drawing::Imaging::PixelFormat::Format24bppRgb, (System::IntPtr)origimage-
    >imageData);
    //=====
    pictureBox_1->Refresh();
}
}

private: System::Void radioButton_1sisi_CheckedChanged(System::Object^ sender,
System::EventArgs^ e) {
    jenis_cetakan=1;
}
private: System::Void radioButton_2sisi_CheckedChanged(System::Object^ sender,
System::EventArgs^ e) {
    jenis_cetakan=2;
}

// Button cari awal X dan Y file scan
private: System::Void button_cari_awalXY_Click(System::Object^ sender, System::EventArgs^
e) {
    //=====
    if(tampung_erot == 0)
    {
        tampung_erot = cvCreateImage(cvSize(erot->width, erot->height), erot->depth, 1);
    }
    if(tampung_erot != erot)
    {
        cvReleaseImage( &tampung_erot ); // fungsinya menghapus image jika tidak digunakan
        tampung_erot = cvCreateImage(cvSize(erot->width, erot->height), erot->depth, 1);
    }
    tampung_erot=cvCloneImage(erot);
    //=====
    simpanx=10000;
    simpany=10000;
    for(starty=0; starty<tampung_erot->height-40; ++starty)
    {
        for(startx=0; startx<tampung_erot->width-50; ++startx)
        {
            if ((tampung_erot->imageData[tampung_erot-
>widthStep*(starty)+(startx)*tampung_erot->nChannels] == 0) && (startx < simpanx))
            {
                simpanx = startx;
                if (starty < simpany)
                {
                    simpany = starty;
                }
            }
        }
    }
} // akhir button cari awal X dan Y

```

```

// Button segmentasi proses file scan
private: System::Void button_segmentasiScan_Click(System::Object^ sender,
System::EventArgs^ e) {
int m1=0, n1=0, area_X_awal=0, area_Y_awal=0, itung_kode=0;

for( j=simpanx;j<tampung_erot->height-40;j=j+30)
{
    textBox_tampung->Text="";
    for( i=simpanx;i<tampung_erot->width-10;i=i+18)
    {
        while (((tampung_erot->imageData[tampung_erot->widthStep*(j+2)+(i)*tampung_erot-
>nChannels] != 0) &&
(tampung_erot->imageData[tampung_erot->widthStep*(j+17)+(i)*tampung_erot-
>nChannels] != 0) &&
(tampung_erot->imageData[tampung_erot->widthStep*(j+27)+(i)*tampung_erot-
>nChannels] != 0)) &&
((tampung_erot->imageData[tampung_erot->widthStep*(j+2)+(i+9)*tampung_erot-
>nChannels] == 0) ||
(tampung_erot->imageData[tampung_erot->widthStep*(j+17)+(i+9)*tampung_erot-
>nChannels] == 0) ||
(tampung_erot->imageData[tampung_erot->widthStep*(j+27)+(i+9)*tampung_erot-
>nChannels] == 0) ))
        {
            i=i+1;
        };
    }
    j2 = j;
    while ((tampung_erot->imageData[tampung_erot->widthStep*(j)+(i+5)*tampung_erot-
>nChannels] == 0) ||
(tampung_erot->imageData[tampung_erot->widthStep*(j)+(i+15)*tampung_erot-
>nChannels] == 0))
    {
        j = j - 1;
    };

    sudut1.y = j;
    sudut2.y = j + 22;
    sudut1.x = i;
    sudut2.x = i + 12;
    area_X_awal = sudut1.x;
    area_Y_awal = sudut1.y;
    cvRectangle(tampil_1,sudut2,sudut1,CV_RGB(0,0,255),1,8,0);
    //=====

// segmentasi kecil (sebanyak 40 >> 5 kolom dan 8 baris
int segmen_X=4;
int segmen_Y=4;
itung_kode=0;
for (m1=0;m1<=4;m1++)
{
    sudut1.x = area_X_awal + (segmen_X * m1);
    sudut2.x = area_X_awal + (segmen_X * (m1+1));
    for (n1=0;n1<=7;n1++)
    {
        sudut1.y = area_Y_awal + (segmen_Y * n1);
        sudut2.y = area_Y_awal + (segmen_Y * (n1+1));
        cvRectangle(tampil_1,sudut2,sudut1,CV_RGB(255,0,0),1,8,0);
    }
}
//=== Tampilkan image ===
if(tampil_Scan == 0)
{
    tampil_Scan = cvCreateImage( cvGetSize(tampil_1), 8, 3 );
}
if(tampil_Scan != 0)
{
    cvReleaseImage( &tampil_Scan ); // fungsinya menghapus image jika tidak digunakan
    tampil_Scan = cvCreateImage( cvGetSize(tampil_1), 8, 3 );
}

```

```

//=====
pictureBox_1->Image = gcnw
System::Drawing::Bitmap(tampil_1->width,tampil_1->height,tampil_1->widthStep,
System::Drawing::Image::PixelFormat::Format24bppRgb, (System::IntPtr)tampil_1-
>imageData);
pictureBox_1->Refresh();
}; }; } // akhir segmentasi

// Proses ANN file Scan
private: System::Void button_proses_ANN_scan_Click(System::Object^ sender,
System::EventArgs^ e) {
StreamWriter^ tulis_hasil_scan = gcnw StreamWriter ("hasil_text_scan.txt");
//=====
richTextBox_hasil->Text="";
richTextBox_hasil->Text="          INI ADALAH HASIL KONVERSI\r\n";
textBox_huruf->Text="";
this->btn_ambil_bobot_ANN_Click_1(sender, e);          // ambil bobot untuk proses ANN
int
awal_baris=1,awal_kolom=1,bb=0,kk=0,cek_x=0,cek_y=0,itung_kode=0,hasil_baca=0,acuan_x=0,ac
uan_y=0;
int
scanX_awal=0,scanX_akhir=0,scanY_awal=0,scanY_akhir=0,scan_X=0,scan_Y=0,kode_in[50],jumlah
baca;
int
perulangan=0,m1,n1,m2,segmen_X,segmen_Y,area_X_awal,area_Y_awal,area_X_akhir,area_Y_akhir;
int bitke0,bitke1,bitke2,bitke3,bitke4,bitke5;

tanda_angka=0;
kapital=0;
//=====

baris=barisScan;
kolom=kolomScan;
for (bb=awal_baris;bb<baris;bb=bb+3)
{
    for (kk=awal_kolom;kk<kolom;kk=kk+2)
    {
        itung_kode=0;
        for (cek_y=bb;cek_y<bb+3;cek_y++)
        {
            for (cek_x=kk;cek_x<kk+2;cek_x++)
            {
                hasil_baca=0;
                itung_kode++;
                acuan_x=koordinat_xScan[cek_x];
                acuan_y=koordinat_yScan[cek_y];
                //=====
                scanX_awal=acuan_x - 3;
                scanX_akhir=acuan_x + 4;
                scanY_awal=acuan_y - 0;
                scanY_akhir=acuan_y + 1;
                //=====
            }
        }
        if (itung_kode==1){area_X_awal=scanX_awal;area_Y_awal=scanY_awal;sudut1.y = scanY_awal-
1;sudut1.x = scanX_awal-1;};
        if (itung_kode==6)
        {
            area_X_akhir=scanX_akhir;
            area_Y_akhir=scanY_akhir;
            sudut2.y = scanY_akhir+1;
            sudut2.x = scanX_akhir;
        }
    }; //akhir cek_x
} //akhir cek_y

// segmentasi kecil (sebanyak 40 >> 5 kolom dan 8 baris
segmen_X=3;
segmen_Y=2;
itung_kode=0;
for (m1=0;m1<5;m1++)
{
    sudut1.x = area_X_awal + (segmen_X * m1);

```

```

        sudut2.x = area_X_awal + (segmen_X * (m1+1));
        for (n1=0;n1<8;n1++)
        {
            sudut1.y = area_Y_awal + (segmen_Y * n1);
            sudut2.y = area_Y_awal + (segmen_Y * (n1+1));
            if (n1==7) {sudut2.y = sudut2.y +1;};
            //=== pembacaan area tiap segmentasi kecil. apakah hitam atau putih
            hitam=0;putih=0;
            hasil_baca=0;
            itung_kode++;
            for (scan_Y=sudut1.y;scan_Y<=sudut2.y;scan_Y++)
            {
                for (scan_X=sudut1.x;scan_X<=sudut2.x;scan_X++)
                {
                    if (imagecontour_Scan->imageData[imagecontour_Scan->widthStep*scan_Y+
                    scan_X*imagecontour_Scan->nChannels] == 0)
                    {hitam++;}
                    else
                    {putih++;};
                }
                // akhir pembacaan area segmentasi kecil
            }
            if (hitam>2) {hasil_baca=1;};
            //=====
            if (itung_kode==1){kode_in[1]=hasil_baca;};
            if (itung_kode==2){kode_in[3]=hasil_baca;};
            if (itung_kode==3){kode_in[5]=hasil_baca;};
            if (itung_kode==4){kode_in[7]=hasil_baca;};
            if (itung_kode==5){kode_in[9]=hasil_baca;};
            if (itung_kode==6){kode_in[11]=hasil_baca;};
            if (itung_kode==7){kode_in[13]=hasil_baca;};
            if (itung_kode==8){kode_in[15]=hasil_baca;};
            if (itung_kode==9){kode_in[2]=hasil_baca;};
            if (itung_kode==10){kode_in[4]=hasil_baca;};
            if (itung_kode==11){kode_in[6]=hasil_baca;};
            if (itung_kode==12){kode_in[8]=hasil_baca;};
            if (itung_kode==13){kode_in[10]=hasil_baca;};
            if (itung_kode==14){kode_in[12]=hasil_baca;};
            if (itung_kode==15){kode_in[14]=hasil_baca;};
            if (itung_kode==16){kode_in[16]=hasil_baca;};
            if (itung_kode==17){kode_in[17]=hasil_baca;};
            if (itung_kode==18){kode_in[18]=hasil_baca;};
            if (itung_kode==19){kode_in[19]=hasil_baca;};
            if (itung_kode==20){kode_in[20]=hasil_baca;};
            if (itung_kode==21){kode_in[21]=hasil_baca;};
            if (itung_kode==22){kode_in[22]=hasil_baca;};
            if (itung_kode==23){kode_in[23]=hasil_baca;};
            if (itung_kode==24){kode_in[24]=hasil_baca;};
            if (itung_kode==25){kode_in[25]=hasil_baca;};
            if (itung_kode==26){kode_in[27]=hasil_baca;};
            if (itung_kode==27){kode_in[29]=hasil_baca;};
            if (itung_kode==28){kode_in[31]=hasil_baca;};
            if (itung_kode==29){kode_in[33]=hasil_baca;};
            if (itung_kode==30){kode_in[35]=hasil_baca;};
            if (itung_kode==31){kode_in[37]=hasil_baca;};
            if (itung_kode==32){kode_in[39]=hasil_baca;};
            if (itung_kode==33){kode_in[26]=hasil_baca;};
            if (itung_kode==34){kode_in[28]=hasil_baca;};
            if (itung_kode==35){kode_in[30]=hasil_baca;};
            if (itung_kode==36){kode_in[32]=hasil_baca;};
            if (itung_kode==37){kode_in[34]=hasil_baca;};
            if (itung_kode==38){kode_in[36]=hasil_baca;};
            if (itung_kode==39){kode_in[38]=hasil_baca;};
            if (itung_kode==40){kode_in[40]=hasil_baca;};
            //=====
        }
        // akhir scan n1
    }
    // akhir scan m1

//PROSES A N N
//=====
for (m1 = 1; m1 <= node_hidden1; m1++)
{

```



```

        temp=0;
        for (n1 = 1; n1 <= banyak_input; n1++)
        {
            temp = temp + (WL1[m1][n1] * kode_in[n1]);
        }
        temp = temp + BIAS [1][m1];
        HASIL_LAYER[1][m1] = 1/(1+exp(-alpha * temp));
        Y[1][m1] = HASIL_LAYER[1][m1];
    }
    // y = fungsi sigmoid(sigma w*x) untuk hidden layer ke-2
    for (m1 = 1; m1 <= node_hidden2; m1++)
    {
        temp = 0;
        for (n1 = 1; n1 <= node_hidden1; n1++)
        {
            temp = temp + (WL2[m1][n1] * Y[1][n1]);
        }
        temp = temp + BIAS [2][m1];
        HASIL_LAYER[2][m1] = 1/(1+exp(-alpha * temp));
        Y[2][m1] = HASIL_LAYER[2][m1];
    }

    // y = fungsi sigmoid(sigma w*x) untuk hidden2 layer output
    for (m1 = 1; m1 <= node_output; m1++)
    {
        temp = 0;
        for (n1 = 1; n1 <= node_hidden2; n1++)
        {
            temp = temp + (WL3[m1][n1] * Y[2][n1]);
        }
        temp = temp + BIAS [3][m1];
        HASIL_LAYER[3][m1] = 1/(1+exp(-alpha * temp));
        Y[3][m1] = HASIL_LAYER[3][m1];
    }
    //=====
    bitke0=0;bitke1=0;bitke2=0;bitke3=0;bitke4=0;bitke5=0;
    if (Y[3][1] >= thresx){bitke0=1;};
    if (Y[3][2] >= thresx){bitke1=1;};
    if (Y[3][3] >= thresx){bitke2=1;};
    if (Y[3][4] >= thresx){bitke3=1;};
    if (Y[3][5] >= thresx){bitke4=1;};
    if (Y[3][6] >= thresx){bitke5=1;};
    //====
    jumlahbaca=0;
    if (bitke0==1){jumlahbaca=jumlahbaca+1;}
    if (bitke1==1){jumlahbaca=jumlahbaca+2;}
    if (bitke2==1){jumlahbaca=jumlahbaca+4;}
    if (bitke3==1){jumlahbaca=jumlahbaca+8;}
    if (bitke4==1){jumlahbaca=jumlahbaca+16;}
    if (bitke5==1){jumlahbaca=jumlahbaca+32;}
    //=====
    // Spasi
    if (jumlahbaca==0){textBox_huruf->Text=" ";tanda_angka=0;}
    // Huruf A
    if (jumlahbaca==1){if (tanda_angka==1){textBox_huruf->Text="1";}
    else {if (kapital==1){textBox_huruf->Text="A";kapital=0;}
    else {textBox_huruf->Text="a";}
    };};
    // Huruf B
    if (jumlahbaca==2){
    if (tanda_angka==1){textBox_huruf->Text="2";}
    else {if (kapital==1){textBox_huruf->Text="B";kapital=0;}
    else {textBox_huruf->Text="b";} }
    }
    // Huruf C
    if (jumlahbaca==3){
    if (tanda_angka==1){textBox_huruf->Text="3";}
    else {if (kapital==1){textBox_huruf->Text="C";kapital=0;}
    else {textBox_huruf->Text="c";} }
    }
    // Huruf D
    if (jumlahbaca==4){
    if (tanda_angka==1){textBox_huruf->Text="4";}
    else {if (kapital==1){textBox_huruf->Text="D";kapital=0;}

```

```

else {textBox_huruf->Text="d";}} }
// Huruf E
if (jumlahbaca==5){
if (tanda_angka==1){textBox_huruf->Text="5";}
else {if (kapital==1){textBox_huruf->Text="E";kapital=0;}
else {textBox_huruf->Text="e";}} }
// Huruf F
if (jumlahbaca==6) {
if (tanda_angka==1){textBox_huruf->Text="6";}
else {if (kapital==1){textBox_huruf->Text="F";kapital=0;}
else {textBox_huruf->Text="f";}} }
// Huruf G
if (jumlahbaca==7) {
if (tanda_angka==1){textBox_huruf->Text="7";}
else {if (kapital==1){textBox_huruf->Text="G";kapital=0;}
else {textBox_huruf->Text="g";}} }
// Huruf H
if (jumlahbaca==8) {
if (tanda_angka==1){textBox_huruf->Text="8";}
else {if (kapital==1){textBox_huruf->Text="H";kapital=0;}
else {textBox_huruf->Text="h";}} }
// Huruf I
if (jumlahbaca==9){
if (tanda_angka==1){textBox_huruf->Text="9";}
else {if (kapital==1){textBox_huruf->Text="I";kapital=0;}
else {textBox_huruf->Text="i";}} }
// Huruf J
if (jumlahbaca==10){
if (tanda_angka==1){textBox_huruf->Text="0";}
else {if (kapital==1){textBox_huruf->Text="J";kapital=0;}
else {textBox_huruf->Text="j";}} }
// Huruf K
if (jumlahbaca==11){
if (kapital==1){textBox_huruf->Text="K";kapital=0;}
else {textBox_huruf->Text="k";}} }
// Huruf L
if (jumlahbaca==12){
if (kapital==1){textBox_huruf->Text="L";kapital=0;}
else {textBox_huruf->Text="l";}} }
// Huruf M
if (jumlahbaca==13){
if (kapital==1){textBox_huruf->Text="M";kapital=0;}
else {textBox_huruf->Text="m";}} }
// Huruf N
if (jumlahbaca==14){
if (kapital==1){textBox_huruf->Text="N";kapital=0;}
else {textBox_huruf->Text="n";}} }
// Huruf O
if (jumlahbaca==15) {
if (kapital==1){textBox_huruf->Text="O";kapital=0;}
else {textBox_huruf->Text="o";}} }
// Huruf P
if (jumlahbaca==16) {
if (kapital==1){textBox_huruf->Text="P";kapital=0;}
else {textBox_huruf->Text="p";}} }
// Huruf Q
if (jumlahbaca==17) {
if (kapital==1){textBox_huruf->Text="Q";kapital=0;}
else {textBox_huruf->Text="q";}} }
// Huruf R
if (jumlahbaca==18){
if (kapital==1){textBox_huruf->Text="R";kapital=0;}
else {textBox_huruf->Text="r";}} }
// Huruf S
if (jumlahbaca==19) {
if (kapital==1){textBox_huruf->Text="S";kapital=0;}
else {textBox_huruf->Text="s";}} }
// Huruf T
if (jumlahbaca==20) {
if (kapital==1){textBox_huruf->Text="T";kapital=0;}
else {textBox_huruf->Text="t";}} }

```

```

// Huruf U
if (jumlahbaca==21) {
if (kapital==1){textBox_huruf->Text="U";kapital=0;}
else {textBox_huruf->Text="u";} }
// Huruf V
if (jumlahbaca==22) {
if (kapital==1){textBox_huruf->Text="V";kapital=0;}
else {textBox_huruf->Text="v";} }
// Huruf W
if (jumlahbaca==23) {
if (kapital==1){textBox_huruf->Text="W";kapital=0;}
else {textBox_huruf->Text="w";} }
// Huruf X
if (jumlahbaca==24){
if (kapital==1){textBox_huruf->Text="X";kapital=0;}
else {textBox_huruf->Text="x";} }

// Huruf Y
if (jumlahbaca==25) {
if (kapital==1){textBox_huruf->Text="Y";kapital=0;}
else {textBox_huruf->Text="y";} }
// Huruf Z
if (jumlahbaca==26) {
if (kapital==1){textBox_huruf->Text="Z";kapital=0;}
else {textBox_huruf->Text="z";} }
// Kapital
if (jumlahbaca==27){kapital=1;}
// tanda angka
if (jumlahbaca==28){tanda_angka=1;}
// Titik (.)
if (jumlahbaca==29){textBox_huruf->Text=".";}
// Tanda tanya (?)
if (jumlahbaca==30){textBox_huruf->Text="?";tanda_angka=0;kapital=0;}
// Koma
if (jumlahbaca==31){textBox_huruf->Text=",";}
// Tanda Seru
if (jumlahbaca==32){textBox_huruf->Text="!";}
// Tanda Strip -
if (jumlahbaca==33){textBox_huruf->Text="-";}
// Karakter lainnya
if (jumlahbaca>33){textBox_huruf->Text=" ";} // diisi spasi

cvWaitKey(50);
textBox_tampung->Text=textBox_tampung->Text+textBox_huruf->Text;
if(textBox_huruf->Text==" "){textBox_huruf->Text=" "+textBox_huruf->Text;};
if(textBox_huruf->Text==","){textBox_huruf->Text=textBox_huruf->Text+",";};
if(textBox_huruf->Text==" "){textBox_huruf->Text=" "+textBox_huruf->Text;};
textBox_huruf->Text="";
//=====
hasil_baca=0;
itung_kode=0;
} // akhir scan kolom
if (perulangan==1) {tuliskan_hasil_1->Write(textBox_tampung->Text+"\r\n");}
if (perulangan==2) {tuliskan_hasil_2->Write(textBox_tampung->Text+"\r\n");}
if (perulangan==3) {tuliskan_hasil_3->Write(textBox_tampung->Text+"\r\n");}
if (perulangan==4) {tuliskan_hasil_4->Write(textBox_tampung->Text+"\r\n");}
if (perulangan==5) {tuliskan_hasil_5->Write(textBox_tampung->Text+"\r\n");}
if (perulangan==6) {tuliskan_hasil_6->Write(textBox_tampung->Text+"\r\n");}
//=====
textBox_tampung->Text="";
} // akhir scan baris
}; // akhir perulangan

tuliskan_hasil_1->Close();
tuliskan_hasil_2->Close();
tuliskan_hasil_3->Close();
tuliskan_hasil_4->Close();
tuliskan_hasil_5->Close();
tuliskan_hasil_6->Close();
//=====

```

```

// Ambil data teks hasil pembacaan
// dan digabungkan menjadi satu
// di tampilkan di richTextBox_hasil
// variabel tampung hasil teks 1 per baris
System::String^ hasil_teks_11;
System::String^ hasil_teks_12;
System::String^ hasil_teks_13;
System::String^ hasil_teks_14;
System::String^ hasil_teks_15;
System::String^ hasil_teks_16;
System::String^ hasil_teks_17;
System::String^ hasil_teks_18;
System::String^ hasil_teks_19;

// variabel tampung hasil teks 2 per baris
System::String^ hasil_teks_21;
System::String^ hasil_teks_22;
System::String^ hasil_teks_23;
System::String^ hasil_teks_24;
System::String^ hasil_teks_25;
System::String^ hasil_teks_26;
System::String^ hasil_teks_27;
System::String^ hasil_teks_28;
System::String^ hasil_teks_29;

// variabel tampung hasil teks 3 per baris
System::String^ hasil_teks_31;
System::String^ hasil_teks_32;
System::String^ hasil_teks_33;
System::String^ hasil_teks_34;
System::String^ hasil_teks_35;
System::String^ hasil_teks_36;
System::String^ hasil_teks_37;
System::String^ hasil_teks_38;
System::String^ hasil_teks_39;

// variabel tampung hasil teks 4 per baris
System::String^ hasil_teks_41;
System::String^ hasil_teks_42;
System::String^ hasil_teks_43;
System::String^ hasil_teks_44;
System::String^ hasil_teks_45;
System::String^ hasil_teks_46;
System::String^ hasil_teks_47;
System::String^ hasil_teks_48;
System::String^ hasil_teks_49;

// variabel tampung hasil teks 5 per baris
System::String^ hasil_teks_51;
System::String^ hasil_teks_52;
System::String^ hasil_teks_53;
System::String^ hasil_teks_54;
System::String^ hasil_teks_55;
System::String^ hasil_teks_56;
System::String^ hasil_teks_57;
System::String^ hasil_teks_58;
System::String^ hasil_teks_59;

// variabel tampung hasil teks 6 per baris
System::String^ hasil_teks_61;
System::String^ hasil_teks_62;
System::String^ hasil_teks_63;
System::String^ hasil_teks_64;
System::String^ hasil_teks_65;
System::String^ hasil_teks_66;
System::String^ hasil_teks_67;
System::String^ hasil_teks_68;
System::String^ hasil_teks_69;

StreamReader^ baca_file1 = gcnew StreamReader("hasil_text_1.txt");
StreamReader^ baca_file2 = gcnew StreamReader("hasil_text_2.txt");

```

```

StreamReader^ baca_file3 = gcnew StreamReader("hasil_text_3.txt");
StreamReader^ baca_file4 = gcnew StreamReader("hasil_text_4.txt");
StreamReader^ baca_file5 = gcnew StreamReader("hasil_text_5.txt");
StreamReader^ baca_file6 = gcnew StreamReader("hasil_text_6.txt");
//=====
hasil_teks_11 = baca_file1->ReadLine();
hasil_teks_12 = baca_file1->ReadLine();
hasil_teks_13 = baca_file1->ReadLine();
hasil_teks_14 = baca_file1->ReadLine();
hasil_teks_15 = baca_file1->ReadLine();
hasil_teks_16 = baca_file1->ReadLine();
hasil_teks_17 = baca_file1->ReadLine();
hasil_teks_18 = baca_file1->ReadLine();
hasil_teks_19 = baca_file1->ReadLine();
baca_file1->Close();
//=====
hasil_teks_21 = baca_file2->ReadLine();
hasil_teks_22 = baca_file2->ReadLine();
hasil_teks_23 = baca_file2->ReadLine();
hasil_teks_24 = baca_file2->ReadLine();
hasil_teks_25 = baca_file2->ReadLine();
hasil_teks_26 = baca_file2->ReadLine();
hasil_teks_27 = baca_file2->ReadLine();
hasil_teks_28 = baca_file2->ReadLine();
hasil_teks_29 = baca_file2->ReadLine();
baca_file2->Close();

//=====
hasil_teks_31 = baca_file3->ReadLine();
hasil_teks_32 = baca_file3->ReadLine();
hasil_teks_33 = baca_file3->ReadLine();
hasil_teks_34 = baca_file3->ReadLine();
hasil_teks_35 = baca_file3->ReadLine();
hasil_teks_36 = baca_file3->ReadLine();
hasil_teks_37 = baca_file3->ReadLine();
hasil_teks_38 = baca_file3->ReadLine();
hasil_teks_39 = baca_file3->ReadLine();
baca_file3->Close();

//=====
hasil_teks_41 = baca_file4->ReadLine();
hasil_teks_42 = baca_file4->ReadLine();
hasil_teks_43 = baca_file4->ReadLine();
hasil_teks_44 = baca_file4->ReadLine();
hasil_teks_45 = baca_file4->ReadLine();
hasil_teks_46 = baca_file4->ReadLine();
hasil_teks_47 = baca_file4->ReadLine();
hasil_teks_48 = baca_file4->ReadLine();
hasil_teks_49 = baca_file4->ReadLine();
baca_file4->Close();

//=====
hasil_teks_51 = baca_file5->ReadLine();
hasil_teks_52 = baca_file5->ReadLine();
hasil_teks_53 = baca_file5->ReadLine();
hasil_teks_54 = baca_file5->ReadLine();
hasil_teks_55 = baca_file5->ReadLine();
hasil_teks_56 = baca_file5->ReadLine();
hasil_teks_57 = baca_file5->ReadLine();
hasil_teks_58 = baca_file5->ReadLine();
hasil_teks_59 = baca_file5->ReadLine();
baca_file5->Close();

//=====
hasil_teks_61 = baca_file6->ReadLine();
hasil_teks_62 = baca_file6->ReadLine();
hasil_teks_63 = baca_file6->ReadLine();
hasil_teks_64 = baca_file6->ReadLine();
hasil_teks_65 = baca_file6->ReadLine();
hasil_teks_66 = baca_file6->ReadLine();
hasil_teks_67 = baca_file6->ReadLine();

```

```

hasil_teks_68 = baca_file6->ReadLine();
hasil_teks_69 = baca_file6->ReadLine();
baca_file6->Close();

//=====
richTextBox_hasil->Text=richTextBox_hasil->Text+hasil_teks_11+hasil_teks_21+"\r\n";
richTextBox_hasil->Text=richTextBox_hasil->Text+hasil_teks_12+hasil_teks_22+"\r\n";
richTextBox_hasil->Text=richTextBox_hasil->Text+hasil_teks_13+hasil_teks_23+"\r\n";
richTextBox_hasil->Text=richTextBox_hasil->Text+hasil_teks_14+hasil_teks_24+"\r\n";
richTextBox_hasil->Text=richTextBox_hasil->Text+hasil_teks_15+hasil_teks_25+"\r\n";
richTextBox_hasil->Text=richTextBox_hasil->Text+hasil_teks_16+hasil_teks_26+"\r\n";
richTextBox_hasil->Text=richTextBox_hasil->Text+hasil_teks_17+hasil_teks_27+"\r\n";
richTextBox_hasil->Text=richTextBox_hasil->Text+hasil_teks_18+hasil_teks_28+"\r\n";
//==

richTextBox_hasil->Text=richTextBox_hasil->Text+hasil_teks_31+hasil_teks_29+"\r\n";
richTextBox_hasil->Text=richTextBox_hasil->Text+hasil_teks_32+hasil_teks_41+"\r\n";
richTextBox_hasil->Text=richTextBox_hasil->Text+hasil_teks_33+hasil_teks_42+"\r\n";
richTextBox_hasil->Text=richTextBox_hasil->Text+hasil_teks_34+hasil_teks_43+"\r\n";
richTextBox_hasil->Text=richTextBox_hasil->Text+hasil_teks_35+hasil_teks_44+"\r\n";
richTextBox_hasil->Text=richTextBox_hasil->Text+hasil_teks_36+hasil_teks_45+"\r\n";
richTextBox_hasil->Text=richTextBox_hasil->Text+hasil_teks_37+hasil_teks_46+"\r\n";
richTextBox_hasil->Text=richTextBox_hasil->Text+hasil_teks_38+hasil_teks_47+"\r\n";
richTextBox_hasil->Text=richTextBox_hasil->Text+hasil_teks_39+hasil_teks_48+"\r\n";
//==
richTextBox_hasil->Text=richTextBox_hasil->Text+hasil_teks_51+hasil_teks_49+"\r\n";
richTextBox_hasil->Text=richTextBox_hasil->Text+hasil_teks_52+hasil_teks_61+"\r\n";
richTextBox_hasil->Text=richTextBox_hasil->Text+hasil_teks_53+hasil_teks_62+"\r\n";
richTextBox_hasil->Text=richTextBox_hasil->Text+hasil_teks_54+hasil_teks_63+"\r\n";
richTextBox_hasil->Text=richTextBox_hasil->Text+hasil_teks_55+hasil_teks_64+"\r\n";
richTextBox_hasil->Text=richTextBox_hasil->Text+hasil_teks_56+hasil_teks_65+"\r\n";
richTextBox_hasil->Text=richTextBox_hasil->Text+hasil_teks_57+hasil_teks_66+"\r\n";
richTextBox_hasil->Text=richTextBox_hasil->Text+hasil_teks_58+hasil_teks_67+"\r\n";
richTextBox_hasil->Text=richTextBox_hasil->Text+hasil_teks_59+hasil_teks_68+"\r\n";
//=====
tabControl1->SelectTab("tabPage5");
if (pilih_operasi==1) {this->btn_siap_mikro_Click_1(sender, e);};

} // Akhir button proses ANN
tulis_hasil_scan->Close();
tabControl1->SelectTab("tabPage5");
}

//===== END PROGRAM =====
};
}

```

BIOGRAFI PENULIS



Joko Subur dilahirkan di Nganjuk, 14 Maret 1982. Merupakan anak kelima dari tujuh bersaudara pasangan Bapak Djasmadi dan Ibu Umini. Penulis memulai pendidikan di SD Negeri 5 Sonobekel Tanunganom-Nganjuk, lalu melanjutkan di SLTP Negeri 02 Tanjunganom. Penulis menempuh jenjang pendidikan di SMK Negeri 1 Nganjuk. Pada tahun 2005 penulis mengikuti kuliah vokasional 1 tahun di PIKSI-Elektro

ITS. Pada tahun 2007 diterima di Jurusan Teknik Elektro - Universitas Hang Tuah Surabaya dan menyelesaikan studi tahap sarjana pada tahun 2011. Setelah menyelesaikan studi di tahap sarjana, pada tahun 2013 penulis meneruskan pendidikan di Program Magister Jurusan Teknik Elektro Institut Teknologi Sepuluh Nopember dengan memilih bidang keahlian Elektronika Industri. Sekarang penulis tercatat sebagai dosen pengajar di jurusan Teknik Elektro Universitas Hang Tuah Surabaya.

e-mail : jokosubur67@yahoo.com

Hp.: 081 357 76 6767